8-2010

Performance evaluation of warehouses with automated storage and retrieval technologies.

Xiao Cai
University of Louisville

PERFORMANCE EVALUATION OF WAREHOUSES WITH AUTOMATED STORAGE AND RETRIEVAL TECHNOLOGIES

By

Xiao Cai

B.E., Huazhong University of Science and Technology, 2005

M.S., University of Louisville, 2007

A Dissertation

Submitted to the Faculty of

the Graduate School of the University of Louisville

in Partial Fulfillment of the Requirements for the

Doctor of Philosophy

Department of Industrial Engineering

University of Louisville

Louisville, Kentucky

Augest 2010

PERFORMANCE EVALUATION OF WAREHOUSES WITH AUTOMATED STORAGE AND RETRIEVAL TECHNOLOGIES

By

Xiao Cai

B.E., Huazhong University of Science and Technology, 2005

M.S., University of Louisville, 2007

A Dissertation Approved on

July 8th, 2010

by the following Dissertation Committee:

_____

Dissertation Director

_____

_____

_____

_____

DEDICATION

This dissertation is dedicated to my parents

Mr. Jiang Cai

and

Mrs. Meijuan Zhang

who have given me invaluable educational opportunities.

ACKNOWLEDGMENTS

I would like to thank my major professor, Dr. Sunderesh Heragu, for his guidance and patience. I would also like to thank the other committee members, Dr. Suraj Alexander, Dr. William Biles, Dr. Gerald Evans and Dr. Thomas Riedel, for their comments and assistance over the past four years. I would also like to express my thanks to the Department of Industrial Engineering for all the support they provided me for the past four years. I wish to also acknowledge the love and support of my parents, Jiang Cai and Meijuan Zhang for letting me pursue my education. Lastly, I would like to thank my friends here and in China to support me during these years.

ABSTRACT

PERFORMANCE EVALUATION OF WAREHOUSES WITH AUTOMATED STORAGE AND RETRIEVAL TECHNOLOGIES

Xiao Cai

July 8th, 2010

In this dissertation, we study the performance evaluation of two automated warehouse material handling (MH) technologies - automated storage/retrieval system (AS/RS) and autonomous vehicle storage/retrieval system (AVS/RS). AS/RS is a traditional automated warehouse MH technology and has been used for more than five decades. AVS/RS is a relatively new automated warehouse MH technology and an alternative to AS/RS. There are two possible configurations of AVS/RS: AVS/RS with tier-captive vehicles and AVS/RS with tier-to-tier vehicles. We model the AS/RS and both configurations of the AVS/RS as queueing networks. We analyze and develop approximate algorithms for these network models and use them to estimate performance of the two automated warehouse MH technologies.

Chapter 2 contains two parts. The first part is a brief review of existing papers about AS/RS and AVS/RS. The second part is a methodological review of queueing network theory, which serves as a building block for our study.

In Chapter 3, we model AS/RSs and AVS/RSs with tier-captive vehicles as open queueing networks (OQNs). We show how to analyze OQNs and estimate related performance measures. We then apply an existing OQN analyzer to compare the two MH technologies and answer various design questions.

In Chapter 4 and Chapter 5, we present some efficient algorithms to solve SOQN.

We show how to model AVS/RSs with tier-to-tier vehicles as SOQNs and evaluate performance of these designs in Chapter 6.

AVS/RS is a relatively new automated warehouse design technology. Hence, there are few efficient analytical tools to evaluate performance measures of this technology. We developed some efficient algorithms based on SOQN to quickly and effectively evaluate performance of AVS/RS. Additionally, we present a tool that helps a warehouse designer during the concepting stage to determine the type of MH technology to use, analyze numerous alternate warehouse configurations and select one of these for final implementation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. Automated MH technologies in warehouses

This thesis deals with performance evaluation of two automated *material handling* (MH) technologies applied in warehouses.

Warehouses are increasingly employing automation technologies to control costs, extend capacity and improve service. Central computers typically monitor and control all components and operational devices in automated warehouse operations. Warehouse managers can obtain real time data on operations to facilitate performance measurement and control. Advantages of automated MH technologies include:

- **Reducing operating costs and improving cash flow** by managing the inventory efficiently.
- **Increasing customer satisfaction and enhancing customer loyalty** by fulfilling customer orders without errors.
- **Improving operational efficiency and productivity** by managing processes and resources efficiently.

An example of an integrated warehouse MH system is illustrated in Figure 1.1.

A typical warehouse consists of three areas - reserve, forward and cross-dock. Cross-dock is that section of the warehouse reserved for the transport of goods directly from inbound trucks to outbound trucks. These goods are not stored in the warehouse, even temporarily. In its warehouse in Louisville, Ann Taylor, Inc., uses an extensive network of sortation systems to cross-dock almost 60% of the incoming packages. In

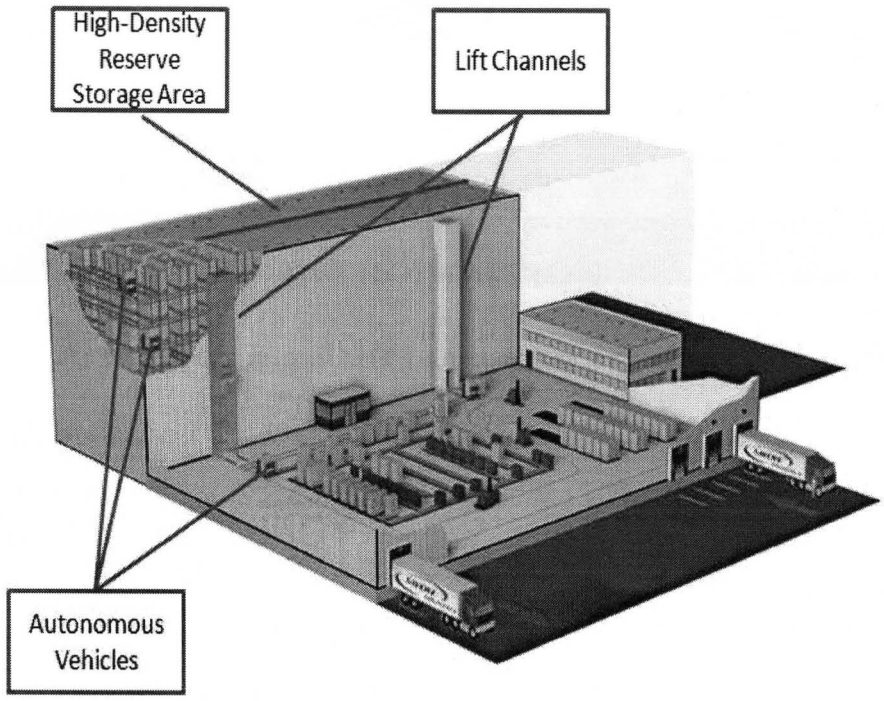

Figure 1.1. An integrated warehouse MH system

the forward section of the warehouse, order-pickers put together customer orders by using items stored in the reserve area or from cartons or pallets delivered directly from the inbound trucks. These two sections together occupy less than 40% of the total warehouse space in a typical warehouse. Over 60% is set aside for the reserve area, which is typically a high density, narrow aisle storage area with pallets stored from wall to wall and floor to ceiling. Because full pallet loads are typically handled in the reserve area and the throughput is high, this section of the warehouse – which has a footprint ranging from 10,000 to 50,000 square feet and height ranging from 50 to 100 feet – is heavily automated.

1.2. Introduction of AS/RS and AVS/RS

In this thesis, we analyze two types of MH technologies used in the reserve area of a warehouse. One of them, called the *automated storage and retrieval system*

(AS/RS), has been widely used for the past five decades. The other, called the *autonomous vehicle storage and retrieval system* (AVS/RS), is relatively new and has been installed in over fifty warehouses in Europe.

Crane-based AS/RS is the technology of choice for *storage/retrieval* (S/R) operations in numerous applications. AS/RS technology has been around for a long time and it can achieve high throughput and fast response times in many MH applications, especially when the throughput is high and stable. In a typical AS/RS, one crane is dedicated to an aisle. Each aisle-captive crane is capable of simultaneous movement in the horizontal and vertical dimensions due to two independent motors. A sample AS/RS that is served by a conveyor system as the input buffer is shown in figure 1.2.

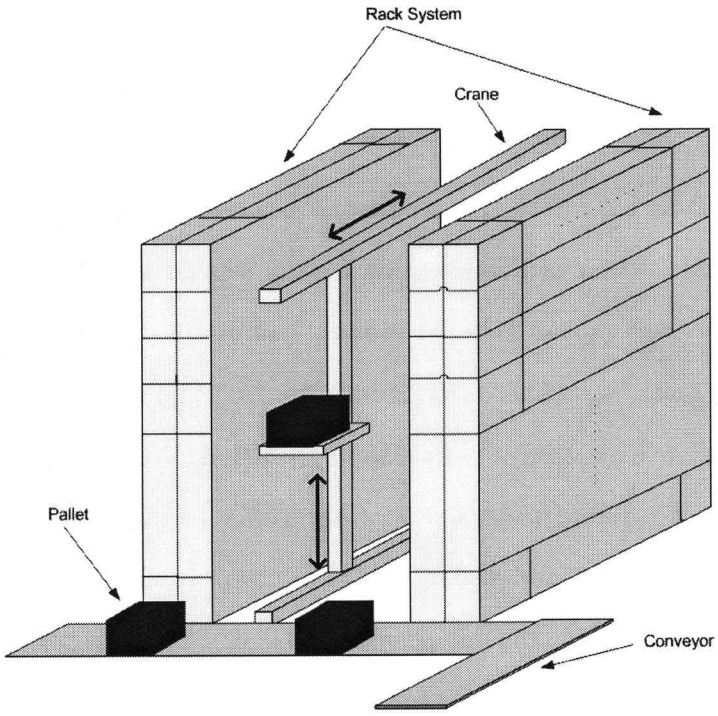

Figure 1.2. A typical AS/RS

Although an AS/RS has numerous advantages, it is a very rigid design and cannot easily adapt to rapid changes encountered in some warehouse operations. The capital

latter increases. An alternative that has been proposed to the AS/RS is the AVS/RS. Developed by Savoye Logistics in France, the AVS/RS has many advantages when compared to an AS/RS in applications requiring flexibility and modularity. The three components of an AVS/RS include:

- autonomous vehicles,
- lifts,
- the rack system.

An AVS/RS is illustrated in figure 1.1. Fork-lift trucks deposit pallets from inbound trucks in a staging area. Rail-guided autonomous vehicles then pick these pallets up and move them to their designated storage position in a designated aisle. If the designated storage position is in another tier, the vehicle interfaces with a lift to reach that tier and uses rails within that tier to travel to the designated storage position. The operations are reversed for a retrieval transaction.

Two main configurations of the AVS/RS are possible.

- AVS/RS with tier-to-tier vehicles
- AVS/RS with tier-captive vehicles.

AVS/RS with tier-captive vehicles operates differently when the designated storage space is in another tier. The vehicle on the ground floor unloads a load in front of the lift, and the lift takes the load to the designated tier. The autonomous vehicle on that tier takes the load from the lift buffer and travels to the assigned storage space. Although all of the AVS/RS installations have tier-to-tier vehicles, it is not difficult for the manufacturer to adapt the design to accommodate a tier-captive configuration.

1.3. Queueing Network Theory

1.3.1. Introduction of queueing network theory

A queueing network can be described as a network consisting of several service *nodes*, and one or more classes of *customers*. The queueing network is used to represent the structure of many complex systems, such as manufacturing systems, computer systems and other systems with a large number of resources. In the general queueing network model, customers belonging to a particular class may enter into the system at any node, complete service there, then transfer to another set of nodes in a specific sequence and leave the system from the last node in the sequence.

We discuss three types of queueing networks. The network described in the preceding paragraph is an *open queueing network* (OQN). Customers in an OQN can arrive into the system at any node and leave the system from any node. A queueing network is called a *closed queueing network* (CQN) when the number of customers is fixed in the system and a customer can neither enter nor leave the system. A queueing network is called a *semi-open queueing network* (SOQN) when each customer needs to be paired with an additional resource before entering the system. This resource stays with the customer until the last stage of service is completed and the customer leaves the system. When that occurs, the resource returns to a pool so it is ready to be paired with the next incoming customer. It should be noted that the OQN implicitly assumes there is an infinite number of these resources, and the CQN assumes there is an infinite number of customers, so one never has to wait for the other. Because that is not always the case, and sometimes a customer may have to wait for the resource or vice versa, the SOQN is a more realistic model of many real-world systems, including the AVS/RS with tier-to-tier vehicles. As mature models, there are many papers about OQNs and CQNs in the literature. Although the SOQN has received relatively

less attention than the OQN and the CQN in the past, this queueing network model is gaining more and more attention now because it has realistic assumptions when compared to the OQN and the CQN.

1.3.2. Queueing network analysis vs discrete event simulation

A straightforward way to evaluate the performance of warehouses is via *discrete event simulation.* The simulation method can be applied to analyze many real systems accurately. Additionally, there are many mature commercial simulation softwares in the market, which can be used to develop simulation models. However, the simulation method has some disadvantages. First, the simulation model development is time consuming. Second, it takes a significant amount of time to run because it needs to be replicated several times to get accurate results. It also consumes computational resources because the simulation model usually contains many of the details of the real system being studied. Third, because a typical simulation model considers many lower level details, a slight change in the real system may lead to a major change in the corresponding simulation model. Additionally, considering many lower level details may distract the designer from seeking the answers to key design questions - the reason why the model was developed in the first place!

Queueing network analysis is an alternative way to evaluate the performance of warehouse configurations. The real system is modeled as a network of queues, and performance measures of this system are estimated by mathematical expressions. There are many advantages of using queueing network analysis. Modeling the real system as a queueing network is relatively simple. Performance evaluation of the corresponding real system can be done quickly. A more important advantage is that the analytical model can analyze alternate system configurations with different sets of input data in an acceptable time and without changing the model itself. Of course,

there are also some disadvantages of queueing network analysis. First, unlike the simulation model, there is no unique modeling framework for queueing networks. Second, the main disadvantage is that the queueing network analysis cannot evaluate complex systems. Therefore, simplification and approximations have to be made to model real systems. Thus, results obtained from the queueing network models are usually approximate. However, ignoring relatively unimportant details allows a faster solution of the model, permitting the evaluation of many more scenarios than is possible via simulation.

A comparison between queueing network analysis and discrete event simulation suggests a two-step design procedure. In the first step, the queueing network analysis could be used to evaluate many different alternatives to find a few candidate designs. In the second step, the simulation method could be used to refine these candidate designs by considering more details.

1.4. Aim and overview

This thesis deals with the performance evaluation of AS/RSs and AVS/RSs. There are many papers on performance evaluation of AS/RSs in the literature. However, performance evaluation of automated warehouses is still mainly done by the simulation method. In this thesis, we develop analytical models to compare AS/RSs and AVS/RSs. We model AS/RSs and AVS/RSs with tier-captive vehicles as OQNs and apply an existing OQN analyzer to evaluate and compare performance of these two technologies. Furthermore, the AVS/RS with tier-to-tier vehicles is a more complicated system and can be only modeled as a multi-class SOQN. We will develop efficient, approximate algorithms to solve this type of SOQN and evaluate the performance of AVS/RSs with tier-to-tier vehicles and compare them with AS/RSs.

This proposal is organized as follows. In Chapter 2, we review the literature on AS/RS and AVS/RS. We also review papers on queueing network theory. In Chapter 3, we model AS/RSs and AVS/RSs with tier-captive vehicles as OQNs. We solve the OQNs to estimate their performance measures. An existing OQN analyzer is used to compare these two technologies and answer various design questions. We model AVS/RSs with tier-to-tier vehicles as SOQNs. These SOQNs are very difficult to analyze; we therefore develop two approximate algorithms to estimate single-class SOQNs with exponentially distributed service times. These are presented in Chapter 4. In Chapter 5, we consider a complex, multi-class SOQN with generally distributed service times. This model is analyzed using an efficient approximate method. Additionally, we propose to develop a software package that embeds all the analytical models we develop for estimating the performance measures of AS/RSs and AVS/RSs and allows a designer to consider numerous, alternative designs in the concepting stage.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

This chapter consists of two parts. The first part is a review of two main automated warehouse technologies discussed in this thesis - AS/RS and AVS/RS. As mentioned in Chapter 1, AS/RS has been used for more than five decades in warehouses. There are many papers in the literature about system modeling, estimation of performance measures and applications of this mature technology. On the other hand, as a relatively new automated MH technology, there are few papers about AVS/RS in the literature. We briefly review existing papers about this technology.

The second part of this chapter is a methodological review of queueing network theory, which can be separated into three sub topics: OQN, CQN and SOQN. To study AS/RS and AVS/RS with tier-captive vehicles, we apply the OQN. To study AVS/RS with tier-to-tier vehicles, we apply the SOQN. Although we do not model the AVS/RS as a CQN directly, the analysis of SOQN models requires the analysis of underlying CQN models. Thus, we review papers on CQNs as well.

2.2. AS/RS literature review

2.2.1. Simulation models

Various simulation models of AS/RSs are available in literature. These studies evaluate alternate design choices for a given system configuration.

2.2.1.1. Sizing problem. The sizing problem of warehouse design is a combined problem, involving internal layouts, storage policies and many other factors. Rosenblatt and Roll (1984) applied optimization techniques to determine the total cost of warehouse sizing with two decision variables: warehouse capacity and storage policy. They extended this work in Rosenblatt and Roll (1988) by developing a simulation model to measure the relationship between warehouse size and various pertinent parameters. The stochastic nature of the demand and replenishment requires us to determine the warehouse capacity by specifying a *desired service level.* The desired service level indicates the proportion of time the warehouse is able to satisfy the demand from stock. Another term in the Rosenblatt and Roll (1988) paper is the *nominal capacity requirement* (NCR), which refers to the average size of a warehouse calculated based on the average quantity of each item. Rosenblatt and Roll (1988) pointed out that a warehouse with an NCR capacity will provide only a 50% service level. There are various parameters that have an effect on the warehouse size for a given service level. These were studied using a simulation model. The *multi-aisle S/R machine system* (MASS) was studied by Hwang and Ko (1988) to reduce the initial investment of installing an AS/RS. The multi-aisle system can reduce the installation cost dramatically to half the cost of single-aisle S/R machine system as long as the S/R demands are relatively low.

Choi and Shin (1997) described a paint body storage at an automobile assembly plant as an AS/RS to re-sequence vehicles for entry into final assembly. Inman (2003) extended this work by applying the AS/RS in automotive assembly sequences. As mentioned, many automotive assembly plants use an AS/RS to adjust the assembly sequence before final assembly. Some primary drivers of the AS/RS size are the degree of sequence scrambling in body and paint shops, the number of vehicle configurations entering final assembly, and each configuration's penetration. Inman

(2003) presented a simulation model for sizing this post-paint AS/RS by considering the above mentioned factors.

2.2.1.2. Deadlock problem. A large number of studies have discussed classical operational problems such as dwell points, expected cycle-time models and optimizing transporter operations (Berg and Gademann (2000)). However, there are relatively very few papers in the literature dealing with operational controller tasks such as avoiding vehicle deadlocks in the AS/RS. Deadlock in manufacturing systems is highly unfavorable because a part's access to resources is held up by other parts. One method of deadlock resolution is to abort one or more parts involved in the deadlock and release the resources to other parts (Fanti et al. (1997)). The approaches to address deadlocks are prevention methods, detection/recovery approaches and avoidance algorithms.

Lee et al. (1996) discussed a deadlock problem in a narrow aisle AS/RS serviced by rail-guided vehicles. They solved the deadlock problem by increasing conveyor capacity in the simulation model, which is a *deadlock detection/recovery* (DDR) method. However, the solution of deadlock problems had not been formally described. In order to characterize the deadlock in the AS/RS correctly, a model should be established. Dotoli and Fanti (2005) suggested a unified modeling framework for the heterogeneous AS/RS transport system by using *colored timed Petri nets* (CTPNs). The CTPN can describe the dynamic behavior of the system, which is modular and resource-oriented. Although the CTPN is resource-oriented and suitable to use at an operational level, it is too complicated for characterizing the deadlock and defining efficient resolution policies for AS/RSs. Dotoli and Fanti (2007) presented their extended work of deadlock detection and avoidance strategies in AS/RSs. The AS/RS is modeled as a timed

discrete event dynamical system (DEDS), in which the information of paths and locations of vehicles is stored in a state. The state can be changed whenever an event occurs. This characterization can be used in the analysis of deadlocks in the AS/RS.

Dotoli et al. (2004) compared two different real-time deadlock solution strategies for the AS/RS: a deadlock avoidance strategy and a deadlock detection/recovery strategy. The deadlock avoidance strategy was proposed by Fanti (2002) to guarantee an efficient system performance using a DEDS and digraph tools. The DDR strategy was proposed by Lee et al. (1996) to solve the deadlock problem by utilizing buffers to store deadlocked jobs.

2.2.1.3. Travel time models. The service time for a transaction includes both S/R machine travel time and pickup/deposit time. The pickup/deposit time is typically assumed to be deterministic due to the nature of the S/R machine. The travel time is variable and is thus useful in measuring important performance measures of AS/RS, for example throughput times. For single shuttle AS/RS, S/R machines can perform up to one storage and one retrieval operation as a *dual command* (DC) cycle. However, in multi-shuttle AS/RS with two unit loads, the S/R machine can perform up to two S/R operations in a cycle as a *quadruple command* (QC) cycle. Potrc et al. (2004) presented a simulation model of multi-shuttle AS/RS by using a new heuristic strategy instead of FCFS strategy in single-shuttle AS/RS. This simulation model indicates that multi-shuttle AS/RS has large improvements in travel time when compared with single-shuttle AS/RS. Hu et al. (2005) presented a continuous travel time model for a new type of AS/RS, *split-platform AS/RS* (SP-AS/RS). By introducing a new S/R mechanism for handling extra heavy loads efficiently, the SP-AS/RS shows improved S/R machine travel times.

2.2.2. Analytical models

Various analytical models to estimate the cost of an AS/RS have been proposed. Once an AS/RS is installed, the performance of the entire system depends on control methods applied on the system. The control methods include storage method, order sequencing, and dwell points of S/R machines. Research in the area of dwell points and expected travel-time models for S/R machines are reviewed in this section.

2.2.2.1. Dwell points models. The first area relates to dwell points. Some simple rule-of-thumb policies have been studied by Bozer and White (1984). These strategies are easy to understand and implement, but they are static and cannot respond to changes in the S/R transactions in an AS/RS from period to period. Egbelu (1991) developed a dynamic optimal location strategy based on mathematical programming including two separated sub models to minimize the service response time. One sub model is to minimize the maximum S/R machine response time while the other solves the minimization of the expected response time. Hwang and Lim (1993) extended the Egbelu (1991) study by transforming one sub model into a single-facility location problem and developed an efficient algorithm to generate an optimal dwell point of S/R machines in AS/RS. Several dwell point specification strategies for S/R machines have been studied by Egbelu and Wu (1993) based on the simple rule-of-thumb policies by Bozer and White (1984) and dynamic strategies based on linear program developed by Egbelu (1991). Egbelu and Wu (1993) conducted a performance comparison by using average order turnaround time as the basis for comparison. The choice of the dwell point has a significant impact on expected response time of AS/RS. Peters et al. (1996) developed an analytical model for the determination of the optimal dwell point location for an S/R machine. This model provides a closed form solution for the dwell point location problem under a variety of system configurations. Peters et al. (1996) developed a closed form solution for *square-in-time* (SIT) racks. However, racks are

not necessarily SIT, which means study of *non-square-in-time* (NSIT) and uniformly distributed racks are more valuable. Park (2001) developed a closed form solution for the optimal dwell point of NSIT racks determined by the probability of the next transaction demand type – storage or retrieval. In addition, various return paths to dwell points are also examined in this paper.

2.2.2.2. Expected travel time models. Another area of study in the literature is the expected travel time of S/R machines in AS/RS. Three storage assignment rules have been compared based on expected travel time of S/R machines by Hausman et al. (1976). They pointed out that there is a significant reduction in crane travel time by using dedicated storage policy such as full turnover-based assignment rather than randomized storage policy. Graves et al. (1977) extended the work done by Hausman et al. (1976) to compare the operating performance of several storage assignment policies by using both continuous and discrete evaluation models. Each rule is compared on the basis of expected travel time of S/R machines. Bozer and White (1984) developed travel time models for single - and dual - command cycles. They compared the expected travel time of an AS/RS crane for these two cycles. Travel times under different storage assignments have been investigated by Wen et al. (2001). They considered various travel speeds with known acceleration and deceleration rates. A computerized algorithm developed by Mansuri (1997) investigated dedicated storage allocation alternatives for an AS/RS based on cycle time of the S/R crane. Ashayeri et al. (2002) presented an exact, geometry-based analytical model to compute the expected cycle travel time for an S/R machine with single-command, dual-command, or both. The rack can be either SIT or NSIT and no fixed layout shape is assumed in this model. This approach can make the AS/RS more appealing for use in integrated supply chain systems. Sari et al. (2007) presented closed-form, travel time expressions for a flow-rack AS/RS based on a continuous approach and compared them

with simulation to demonstrate that this analytical model can estimate performance measures by requiring less computing time than simulation. Duc and De Koster (2007) developed an optimization method to determine the average throughput time for an order batching problem in a 2-block rectangular warehouse by applying the well-known S-shape heuristic method.

2.2.2.3. Dynamic control policies. Proper selection of dynamic control policies allows us to maximize the system throughput. Lin and Wang (1995) presented an application of *stochastic Petri nets* (SPNs) to describe the behavior of AS/RS and evaluate the performance of different control policies of such systems. The SPN is a graph-based tool that can build a system at different levels. This property of SPN can divide an entire system into several sub systems and model them separately, which makes it is easy to remodel changes in system configurations. As mentioned previously, Dotoli and Fanti (2005) developed a *colored Petri net* (CPN) to investigate the performance of AS/RS from a control perspective. CPN is a well-known dialect of high-level SPNs that can be implemented in work flow analysis. This allows us to model a resource-oriented model suitable for real-time control. A performance analysis for multiple aisle AS/RS by using SPNs is presented by Benamar et al. (2003). *Timed Petri nets* (TPNs) extended from CPNs with time concepts is applied to model the AS/RS and evaluate system performance. Furmans et al. (2008) showed the impact of batch building on the throughput times and resource consumption.

2.3. AVS/RS literature review

AVS/RS is a relatively new automated warehouse MH technology, which was first introduced to the U.S. in 2000 (Malmborg (2002)). Hence, there are only a few papers on the analysis of AVS/RS. These papers present analytical models to estimate performance measures of the AVS/RS.

Malmborg (2002) developed a conceptualization tool to analyze the effects of rack configuration, storage capacity and configuration of autonomous devices in system performance of AVS/RS.

Some of the following papers discuss the estimation of cycle times for autonomous devices in an AVS/RS. Malmborg (2003) proposed an estimation model based on Markov Chains. Fukunari and Malmborg (2007) analyzed the expected travel time of autonomous vehicles based on vehicle movements. Kuo et al. (2007) modeled the movement of autonomous devices as an $M/G/V$ queue nested within an $M/G/L$ queue for estimating the expected travel times of vehicles and lifts. Vehicle movements are treated as service nodes of $M/G/V$ queues and lift movements are treated as service nodes of $M/G/L$ queues. An AVS/RS with a *point of service completion* (POSC) dwell point policy is discussed in Kuo et al. (2006). Fukunari et al. (2004) discussed dwell point issues of AVS/RS in detail by using a decision-tree analysis.

Zhang (2008) proposed a methodological foundation for design conceptualization of AVS/RS. Special queueing network models have been developed to compare AVS/RSs and AS/RSs to give insights for the selection of automated warehouse technologies based on requirements of throughput rates, storage capacity and system configuration.

Ekren et al. (2010) discussed how to use simulation models to find key factors to affect performance of an AVS/RS.

2.4. Introduction of queueing network theory

In this section, we briefly discuss queueing networks. We also present an important set of results relative to the queueing network theory.

2.4.1. Queuing networks notation and performance measures

The following symbols are used in the description of queueing networks:

M Number of service nodes

C Number of customer classes

m_i Number of parallel servers at the ith node ($m_i \geq 1$)

μ_{ic} Service rate at ith node for cth class of customers

k_{ic} Number of cth class of customers at ith node

k_i Number of customers at ith node

$$k_i = \sum_{c=1}^{C} k_{ic} \tag{2.1}$$

$p_{ic,js}$ The probability that a customer of cth class at the ith node is transferred to the jth node as sth class

$p_{0,ic}$ The probability that a cth class customer enters the system from outside at ith node

$p_{ic,0}$ The probability that a cth class customer leaves the system at ith node

$$p_{ic,0} = 1 - \sum_{j=1}^{M} \sum_{s=1}^{C} p_{ic,js} \tag{2.2}$$

λ The overall arrival rate

$\lambda_{0,ic}$ The arrival rate from outside to ith node for cth class of customers

λ_{ic} The arrival rate of cth class of customers at ith node

$$\lambda_{ic} = \lambda p_{0,ic} + \sum_{j=1}^{M} \sum_{s=1}^{C} \lambda_{js} p_{js,ic} \tag{2.3}$$

vr_{ic} The mean number of visits of a cth class customer at the ith node

$$vr_{ic} = p_{0,ic} + \sum_{j=1}^{M} \sum_{s=1}^{C} vr_{js} p_{js,ic} \tag{2.4}$$

Usually $vr_{1c} = 1$

V The number of additional resources in an SOQN

The main performance measures of queueing networks are:

Lq_{ic} The mean number of cth class of customers waiting in front of ith node

L_{ic} The mean number of cth class of customers at ith node

Lq_i The average queue length in front of ith node

L_i The average number of customers in ith node

W_c The average time a cth class of customer spends in the system

L_{eq} The average queue length of the external queue in an SOQN

L_{pq} The average number of idle resources in an SOQN

2.4.2. Product-form queueing networks

Generally, queueing networks can be analyzed via a solution of their state spaces. However, some queueing networks with special structures can be analyzed without generating the state space. Queueing networks with these structures are called *product-form* queueing networks. Many OQN and CQN methods are based on some important theoretical results for product-form queueing networks.

First, we introduce the concepts of *global balance* and *local balance*. The behavior of a queueing network can be described by using *continuous time Markov chains* (CTMCs). More details about modeling queueing networks by CTMCs are discussed in Chapter 4. An important result of CTMC is a set of global balance equations as shown in Theorem 1:

Theorem 1. *If the CTMC is ergodic, a unique steady-state probability vector* $\vec{\pi}$ *independent of the initial probability vector exists, and* $\vec{\pi}$ *satisfies:*

$$\vec{\pi}\mathbf{Q} = \mathbf{0}, \tag{2.5}$$

where $\mathbf{Q}$ *is the infinitesimal generator matrix (instantaneous transition rate matrix) of this CTMC.*

We can always solve the global balance equations of a queueing network. However, this method is not suitable for large queueing networks because the number of equations increases exponentially even if the number of nodes increases linearly. Chandy (1972) proved in his paper that the global balance equations can be decomposed into simpler equations or local balance equations if the queueing network satisfies the following property:

Property 1. *The inter-arrival times of all classes of customers and service times at all nodes are exponentially distributed. For each node in this network, the departure rate from a state of the queueing network due to the departure of a customer from ith node equals the arrival rate to this state due to an arrival of a customer at this node. The global balance equations can be split into a number of single equations, each related to an individual node. For generally distributed inter-arrival and service times, the arrival and departure rates should be considered on phases, not on nodes.*

We use a simple example to illustrate the difference between global balance equations and local balance equations.

Example 1. *Consider a CQN with three nodes as shown in figure 2.1. Assume there are two customers in the system. The routing probabilities are:* $p_{12} = p_{13} = 0.5$, $p_{21} = p_{31} = 1$. *Assume this queue network satisfies Property 1.*

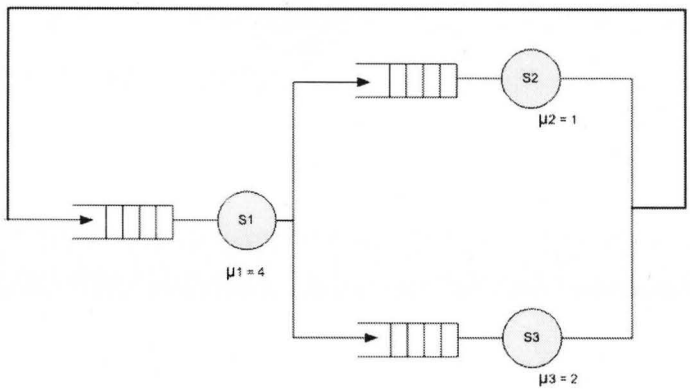

Figure 2.1. A simple CQN with three nodes

All the possible states can be written as: (2,0,0), (0,2,0), (0,0,2), (1,1,0), (1,0,1), (0,1,1). The CTMC of this CQN is shown in figure 2.2.

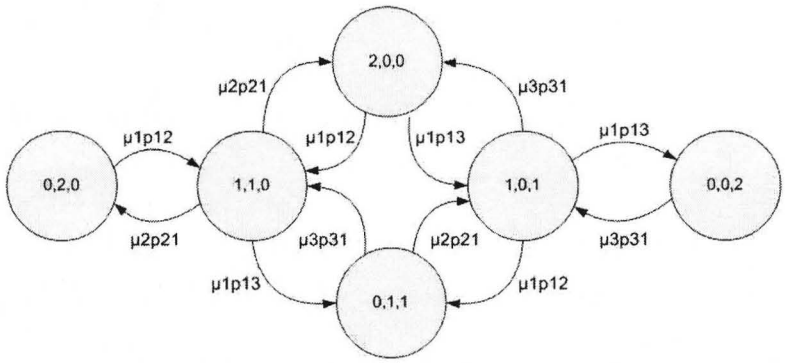

Figure 2.2. CTMC of the CQN in Example 1

We determine one of the global balance equations below:

$$\pi(1,1,0)(\mu_2 p_{21} + \mu_1 p_{13} + \mu_1 p_{12}) = \pi(0,2,0)\mu_2 p_{21} + \pi(2,0,0)\mu_1 p_{12} + \pi(0,1,1)\mu_3 p_{31}.$$

This global balance equation can be split into two local balance equations. According to Property 1, the departure rate from state (1,1,0) due to the departure of a customer from node 2 ($\pi(1,1,0)\mu_2 p_{21}$) equals the arrival rate to state (1,1,0) due to a customer

arrivals to node 2 ($\pi(2,2,0)\mu_1 p_{12}$). This is the first local balance equation:

$$\pi(1,1,0)\mu_2 p_{21} = \pi(2,0,0)\mu_1 p_{12}.$$

For the remaining part of this global equation, the departure rate from state (1,1,0) due to the departure of a customer from node 1 ($\pi(1,1,0)(\mu_1 p_{13} + \mu_1 p_{12})$) equals the arrival rate to this state due to a customer arrivals to node 1 ($\pi(0,2,0)\mu_2 p_{21} + \pi(0,1,1)\mu_3 p_{31}$). The second local balance equation is:

$$\pi(1,1,0)(\mu_1 p_{13} + \mu_1 p_{12}) = \pi(0,2,0)\mu_2 p_{21} + \pi(0,1,1)\mu_3 p_{31}.$$

From this simple example, we can see global balance equations can be reduced to simpler local balance equations. However, not all queueing networks hold Property 1. We say queueing networks that satisfy Property 1 have *product-form solutions* for the steady-state probabilities.

There are several famous product-form queueing networks in the literature. We present definitions and important results of these queueing networks below.

2.4.2.1. Jackson networks. The concept of product-form queueing networks first appeared in Jackson (1963). This paper is very important because it introduced the product-form solution of a certain set of OQNs. These OQNs are called *Jackson networks.*

Definition 1. *A Jackson network is a single-class OQN with M serivce nodes. Customers can enter the system from any node and leave the system at any node. All arrival processes are Poisson and all service times are exponentially distributed. The arrival rate λ_{0i} and the service rate μ_i can depend on the number of customers*

k_i, which are called load-dependent arrival rates and load-dependent service rates respectively. Queues in front of all the nodes follow a first-come, first-served (FCFS) discipline.

The important theorem of Jackson networks is shown below.

Theorem 2. *If an OQN satisfies the assumptions in Definition 1, and is stable at all nodes ($\lambda_i < \mu_i m_i$, $i = 1, \cdots, M$), the steady-state probabilities of this OQN can be expressed by the product of steady-state probabilities of individual nodes:*

$$\pi(k_1, k_2, \cdots, k_M) = \pi_1(k_1) \cdot \pi_2(k_2) \cdot \cdots \cdot \pi_M(k_M). \tag{2.6}$$

This theorem is proved by verifying that equation (2.6) fulfills global balance equations (Jackson (1963)). Theorem 2 indicates that Jackson networks can be decomposed into several independent $M/M/m$ queues with arrival rate λ_i and service rate μ_i at ith node. For $M/M/m$ queues, the formula of $\pi_i(k_i)$ is:

$$\pi_i(k_i) = \begin{cases} \pi_i(0)\frac{(m_i\rho_i)^{k_i}}{k_i!}, & k_i \leq m_i, \\ \pi_i(0)\frac{m_i^{m_i}\rho_i^{k_i}}{m_i!}, & k_i > m_i, \end{cases} \tag{2.7}$$

where

$$\pi_i(0) = \left(\sum_{k_i=0}^{m_i-1} \frac{(m_i\rho_i)^{k_i}}{k_i!} + \frac{(m_i\rho_i)^{m_i}}{m_i!(1-\rho_i)}\right)^{-1}, \rho_i = \frac{\lambda_i}{m_i\mu_i}. \tag{2.8}$$

This important thoerem is the foundation of other OQN methodologies.

2.4.2.2. Gordon/Newell networks. The product-form solution of CQN was first introduced by Gordon and Newell (1967). CQNs having a product-form solutions are called *Gordon/Newell networks*.

Definition 2. *A Gordon/Newell network is a single-class CQN with M service nodes and K customers. Customers can neither enter nor leave the network. All*

service times μ_is are exponentially distributed and can be load-dependent. Queues in front of all nodes follow a FCFS discipline.

Theorem 3. *If a CQN satisfies assumptions in Definition 2, the steady-state probabilities of this CQN can be expressed by the following equation:*

$$\pi(k_1, k_2, \cdots, k_M) = \frac{1}{G(K)} \prod_{i=1}^{M} F_i(k_i). \tag{2.9}$$

$G(K)$ is a normalization constant, and it is given by:

$$G(K) = \sum \prod_{i=1}^{M} F_i(k_i). \tag{2.10}$$

$G(K)$ is chosen in such a way that the sum of probabilities over all states in the network equals one. The $F_i(k_i)$s are given by:

$$F_i(k_i) = (\frac{e_i}{\mu_i})^{k_i} \frac{1}{\beta_i(k_i)}, \tag{2.11}$$

where

$$\beta_i(k_i) = \begin{cases} k_i! & k_i \leq m_i, \\ m_i! m_i^{k_i - m_i} & k_i \geq m_i, \\ 1 & m_i = 1. \end{cases} \tag{2.12}$$

If service rates are load-depenent, $F_i(k_i)$ can be generalized as:

$$F_i(k_i) = \frac{e_i^{k_i}}{A_i(k_i)}, \tag{2.13}$$

where

$$A_i(k_i) = \begin{cases} \prod_{j=1}^{k_i} \mu_i(j) & k_i > 0, \\ 1 & k_i = 0. \end{cases} \tag{2.14}$$

Gordon and Newell (1967) proved that equation (2.9) fulfills global balance equations. Thus, Theorem 3 shows a product-form solution of Gordon/Newell networks. Similar to Jackson networks, the steady-state probabilities of Gordon/Newell networks can be expressed by the product of functions $F_i(k_i)$s on individual nodes. This conclusion provides an important foundation for CQN methods we will discuss later.

2.4.3. BCMP networks

An extension of Jackson networks and Gordon/Newell networks was proposed by Baskett et al. (1975). In this classic paper, a larger set of queueing networks with multiple classes of customers, different queueing strategies, and generally distributed service times was proved to have product-form solutions. This set of queueing networks with product-form solutions is called *BCMP networks*.

Definition 3. *A BCMP network is a multi-class queueing network with M service nodes and C classes of customers. The route of each class of customer can be either open or closed. If the route of cth class of customers is closed, the number of customers is k_c. The service discipline at any node can be one of four service disciplines: FCFS, last-come, first-served (LCFS), processor sharing (PS), infinite server (IS). The service time μ_{ic} is generally distributed for LCFS, PS and IS, and exponentially distributed for FCFS.*

Definition 3 indicates that Jackson and Gordon/Newell networks are special cases of BCMP networks. Since there are multiple classes of customers in a BCMP network, the corresponding CTMC is more complicated. In order to describe the state space of this CTMC, the vector notation $\vec{\mathbf{S}} = \{\vec{\mathbf{S}}_1, \cdots, \vec{\mathbf{S}}_M\}$, where $\vec{\mathbf{S}}_i = \{k_{i1} \cdots k_{ic}\}$ is the vector of numbers of different classes of customers at ith node is introduced.

The vector $\vec{\mathbf{K}} = \{\vec{\mathbf{K}}_1, \cdots, \vec{\mathbf{K}}_C\}$ denotes customer numbers of different classes, where $\vec{\mathbf{K}}_c = \sum_{i=1}^{M} k_{ic}$. The product-form solution of BCMP networks is shown below.

Theorem 4. *For BCMP networks, the steady-state probabilities with a product-form solution can be divided into two sets. For a CQN satisfying the assumptions of BCMP:*

$$\pi(\vec{\mathbf{S}}) = \frac{1}{G(\vec{\mathbf{K}})} \prod_{i=1}^{M} F_i(\vec{\mathbf{S}}_i), \tag{2.15}$$

where $G(\vec{\mathbf{K}})$ is defined by equation (2.10), and for an OQN satisfying the assumptions of BCMP:

$$\pi(k_1, \cdots, k_M) = \prod_{i=1}^{M} \pi_i(k_i). \tag{2.16}$$

Some extensions of this important theorem are briefly mentioned here. Bruell and Balbo (1980) extended this theorem by allowing class change in the network. Towsley (1980) and Krzesinski (1987) added *load-dependent routing probabilities* to Theorem 4.

2.5. OQN methodology review

OQN models have been used in many domains like computer science, manufacturing, service centers and other fields. There are numerous papers about the OQN in the literature. Panwalker and Iskander (1977), Graves (1981) and Bitran and Dasu (1992) have written excellent surveys about the OQN methodology. In this section, we first briefly review some famous exact analytical methods for OQN. We then present an important approximation method called the *decomposition method* in detail. After that, some extensions of this method are briefly reviewed.

2.5.1. Exact analysis

OQN model was first introduced by Erlang (1917) in solving the automatic telephone exchange problem. However, this field had not received enough attention until a famous paper by Jackson (1963) was published. This paper introduced the famous Jackson networks, the kind of product-form OQNs we discussed in Section 2. Kelly (1975), Lemoine (1977), Disney and Konig (1985) and Buzacott and Shanthikumar (1985) made some extensions of these OQNs as *reversible networks.*

However, exact analysis can only be applied to product-form OQNs with the following assumptions:

- All nodes in an OQN have exponentially distributed service times.
- Service times for customer classes are independent in multi-class OQNs.
- Arrival process is Poisson.

These restricted assumptions usually are not satisfied in real applications. Therefore, many approximation methods have been developed to solve *non-product-form* OQNs.

2.5.2. Approximate analysis

According to the paper by Bitran and Tirupati (1988), exponentially distributed service times are overestimated in many real applications. Since exact analysis is only valid for exponentially distributed service times, it has been replaced by approximate analysis with looser assumptions.

Besides decomposition methods, there are several other approximation methods in the literature, such as *diffusion approximations* (see details in Kobayashi (1974) and Reiser and Kobayashi (1975)) and *maximum entropy method* (see details in Kouvatsos (1985) and Walstra (1985)).

The decomposition method is utilized in our research. Based on the work from Kuehn (1979), Pujolle and Ai (1986), Whitt (1983) and Gelenbe and Pujolle (1987), this method can be described as follows:

STEP 1: Decompose the original OQN into stochastically independent $GI/G/1$ and $GI/G/m$ queues. Compute the first two moments of the effective service time at each node;

STEP 2: Calculate the effective arrival rate and the coefficient of variation in the inter-arrival times at each node by solving two sets of equations simultaneously;

STEP 3: Compute the mean queue length and other performance measures for each node and aggregate these results to obtain performance measures for the entire network.

Different implementations of these steps lead to different decomposition procedures. One of the most commonly used procedures is the *parametric decomposition* (PD) method. This method was first introduced by Reiser and Kobayashi (1974), and has been modified and developed by Shanthikumar and Buzacott (1981), Whitt (1983), Buzacott and Shanthikumar (1985), Bitran and Tirupati (1988) and Meng et al. (2004).

The main idea of PD is to decompose a complex queuing network into several isolated queues or subsystems (Kuehn (1979)). Key points of PD include two principles. The first one is approximation of all *non-renewal* processes by stationary *renewal* processes. A renewal process means every time an event occurs, the process renews itself and starts all over again. The Poisson process is a special case of a renewal process where time between occurrences is exponentially distributed. Typically, a complex stochastic process has one or more embedded renewal processes, which allows the process to be decomposed into smaller independent systems. For example, Markovian networks can be decomposed into subsystems exactly, while general networks only

can be decomposed approximately. Another important principle is consideration of the first two moments - mean value and *squared coefficient of variation* (SCV) of all processes. This principle rests on a number of observations in queueing networks where mean values of performance measures are mainly influenced by the mean and SCV of a random variable.

The *queueing network analyzer* (QNA) developed by Whitt (1983) is a software package for OQNs in communication systems, but it was modified to model discrete parts manufacturing systems. The approximation method in QNA decomposes the queuing network into several stochastically independent $GI/G/m$ queues. A $GI/G/m$ queue indicates that the arrival process of the queue with m multiple servers is a renewal process with a general distribution (GI), and the distribution of service time is also general (G). QNA uses the first two moments of inter-arrival and service times to handle more generally distributed OQNs. Once the first two moments of the inter-arrival time of each customer type into the network and its routing are given, QNA calculates the first two moments of effective inter-arrival times of customers at each node. QNA also calculates the first two moments of effective service times at each node and then analyzes each node as an independent $GI/G/1$ or $GI/G/m$ queue to estimate performance measures of this node. Finally, performance measures of the entire queuing network are estimated by synthesizing performance measures of these independent $GI/G/1$ or $GI/G/m$ queues (Whitt (1983)).

Here we list some additional notations of QNA:

O_c Number of operations for cth class of customers

λ_{jk} Rate at which a product leaving node j goes to node k

λ'_{jk} Rate at which a product arriving at node k comes from node j

p_{jk} Proportion of products leaving node j that go to node k

p'_{jk} Proportion of products arriving at node k that come from node j

c_{aj}^2 SCV of inter-arrival time for two consecutive arrivals into node j

2.5.2.1. Basic network operations. Whitt (1983) pointed out that an OQN combines several basic network operations no matter how complex the network is. These network operations are *departure*, *split* and *superposition*. Effects of these operations are discussed next.

1) Departure operation

The effective departure rate and SCV of the inter-departure times at a node are:

$$\lambda_d = \lambda_a, \tag{2.17}$$

$$c_d^2 = 1 + (1 - \rho^2)(c_a^2 - 1) + \frac{\rho^2}{\sqrt{m}}(c_s^2 - 1), \tag{2.18}$$

where λ_a is the arrival rate into the node, and c_a^2 is SCV of inter-arrival times. ρ is the traffic intensity or utilization at the node:

$$\rho = \lambda_a \mu_s / m.$$

2) Split operation

The aggregate product is split into subaggregate products via the split operation. The effective departure rate and SCV of the arrival process of split operation are:

$$\lambda_i = p_i \lambda_a, \tag{2.19}$$

$$c_i^2 = p_i c_a^2 + 1 - p_i. \tag{2.20}$$

3) Superposition operation

Products arriving at a node j are aggregated via the superposition operation. The

first two moments of the arrival process of superposition are:

$$\lambda_a = \sum_i \lambda_i, \tag{2.21}$$

$$c_a^2 = \omega \sum_i (\frac{\lambda_i}{\sum_k \lambda_k}) c_i^2 + 1 - \omega. \tag{2.22}$$

The weighting function is

$$\omega = [1 + 2.1(1-\rho)^{1.8}\nu]^{-1},$$

where

$$\nu = [\sum_i (\lambda_i / \sum_k \lambda_k)^2]^{-1}.$$

2.5.2.2. Estimating the first two moments of inter-arrival times. As mentioned previously, the first two moments of effective inter-arrival times are obtained by solving two sytems of linear equations, which synthesize the effects of these three basic network operations on the first and second moments of the inter-arrival times.

The first system of linear equations calculates the effective arrival rate into each node:

$$\hat{\lambda}_k = \hat{\lambda}_{0k} + \sum_{j=1}^{n} \hat{\lambda}_j p_{jk}, \quad k = 1, 2, ...n. \tag{2.23}$$

Equation (2.23) indicates that the effective arrival rate into any node k is equal to the arrival rate from the outside world plus arrival rates from other nodes in the network.

The second system of linear equations is used to caculate the SCV of the inter-arrival time for two consecutive arrivals into any node in the network, which is the synthesis of the effects of the three basic network operations:

$$c_{aj}^2 = a_j + \sum_{i=1}^{n} c_{ai}^2 b_{ij}, \quad 1 \leq j \leq n. \tag{2.24}$$

In equation (2.24), a_j and b_{ij} are constants obtained via the following equations:

$$a_j = 1 + \omega_j((p'_{0j}c^2_{0j} - 1) + \sum p'_{ij}[(1 - p_{ij}) + p_{ij}\rho^2_i x_i]),$$

$$b_{ij} = \omega_j p'_{ij} p_{ij}(1 - \rho^2_i),$$

$$x_i = 1 + m_i^{-0.5}(\max(c^2_{si}, 0.2) - 1),$$

$$\omega_j = [1 + 4(1 - \rho_j)^2(v_j - 1)]^{-1},$$

where

$$v_j = (\sum_{i=0}^{n} p'^2_{ij})^{-1}.$$

2.5.2.3. Estimating the first two moments of service times. The effective mean value and SCV of service times are calculated as:

$$\mu_j = \frac{\sum_{k=1}^{r}\sum_{\ell=1}^{n_k}\lambda_k\mu_{k\ell}1\{(k,\ell) : n_{k\ell} = j\}}{\sum_{k=1}^{r}\sum_{\ell=1}^{n_k}\lambda_k 1\{(k,\ell) : n_{k\ell} = j\}}, \tag{2.25}$$

$$\mu_j^2(c^2_{sj} + 1) = \frac{\sum_{k=1}^{r}\sum_{\ell=1}^{n_k}\lambda_k\mu^2_{k\ell}(c^2_{sk\ell} + 1)1\{(k,\ell) : n_{k\ell} = j\}}{\sum_{k=1}^{r}\sum_{\ell=1}^{n_k}\lambda_k 1\{(k,\ell) : n_{k\ell} = j\}}. \tag{2.26}$$

2.5.2.4. Network performance measures. Now that the network has been broken up into stochastically independent nodes and their effective inter-arrival and service time parameters have been approximately calculated, we can analyze each node separately and obtain estimates of performance measures as shown next.

The average waiting time in a $GI/G/m$ queue is:

$$\mathbb{E}(WQ_k) = \frac{c^2_{ak} + c^2_{sk}}{2}\mathbb{E}(WQ_k)^{M/M/m}, \tag{2.27}$$

where $\mathbb{E}(WQ_k)^{M/M/m}$ is the average waiting time in a $M/M/m$ queue.

The basic total network performance measure is the throughput, which is same as the total external arrival rate λ_0, and

$$\lambda_0 = \sum^{n} \lambda_{0i}. \tag{2.28}$$

The mean and variance of the number of customers N in the entire network are

$$\mathbb{E}N = \sum_{i}^{n} \mathbb{E}N_i, \tag{2.29}$$

$$Var(N) = \sum_{i}^{n} Var(N_i). \tag{2.30}$$

There are two kinds of customers: an aggregate customer and a particular customer. For an aggregate customer, p_{ij} is independent of the current state and history of the network. On the other hand, each particular customer should have a relatively negligible effect on the total network from the view of particular customers.

Some important equations for estimating key performance measures are listed here.

The expected number of visits to node i is

$$\mathbb{E}V_i = \lambda_i/\lambda_0. \tag{2.31}$$

The mean time a customer spends in node i is

$$\mathbb{E}T_i = (\mathbb{E}V_i)(\tau_i + \mathbb{E}W_i). \tag{2.32}$$

The expected total time in the network from arrival to departure for a customer is

$$\mathbb{E}T = \sum_{i=1}^{n} \mathbb{E}T_i = \sum_{i=1}^{n} \mathbb{E}V_i(\tau_i + \mathbb{E}W_i). \tag{2.33}$$

The variance of the time spent by a customer at node i is

$$Var(T_i) = \mathbb{E}V_i(Var(W_i) + \mu_i^2 c_{si}^2) + Var(V_i)(\mathbb{E}W_i + \mu_i)^2, \tag{2.34}$$

and

$$\mathbb{E}V_i^2 = \sum_{j=1}^{n} (\lambda_{0j}/\lambda_0)[F(2F_{dg} - 1)]_{ji}, \tag{2.35}$$

where F is the matrix $(I - P)^{-1}$, and $P \equiv (p_{ij})$. F_{dg} is the $n \times n$ matrix with all off-diagonal entries 0 and diagonal entries the same as F.

By assuming that T_is at the different nodes are conditionally independent,

$$T = (\sum_{j=1}^{n} \sum_{k=1}^{V_j} T_{kj}), \tag{2.36}$$

where T_{kj} is the time for the kth visit to node j.

Finally,

$$\mathbb{E}(T^2) = \sum_{i=1}^{n} \mathbb{E}(\sum_{k=1}^{V_i} T_{ki})^2 + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathbb{E}(\sum_{k=1}^{V_i} T_{ki} \sum_{\ell=1}^{V_j} T_{\ell j}), \tag{2.37}$$

and

$$Var(T) = \mathbb{E}(T^2) - (\mathbb{E}(T))^2. \tag{2.38}$$

Hence,

$$Var(T) = \sum_{i=1} nVar(T_i) + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathbb{E}(T_{1i}\mathbb{E}(T_{1j})Cov(V_i, V_j)). \tag{2.39}$$

The QNA has been applied in different research areas, such as computer science, communication, and semiconductor industry (see Magarajan et al. (1991) and Glassey and Resende (1988)). Yu and Koster (2008) used QNA to analyze pick-and-pass order picking system by modeling the system as a $G/G/m$ queueing network.

Many extensions have been developed based on the QNA. Segal and Whitt (1989) published a queueing network analyzer that extends the QNA by including additonal operations for manufacturing, such as partial yields and tests, lot sizes and service interruptions. Lambrecht et al. (1998) introduced a procedure to analyze a jobshop for multiple products, and modeled this procedure as an OQN and then analyzed the procedure based on the QNA. The extension of this paper is that it considers setup and batching operations in a jobshop. Meng and Heragu (2004) developed the *manufacturing system performance analyzer* (MPA) as an extension of the QNA. In MPA, the batching operation is considered as the fourth basic network operation. Two sets of linear equations of service times and inter-arrival times are modified by adding the batching operation. MPA is discussed further in Chapter 3.

2.6. CQN methodology review

The study of CQNs can be traced back to the 1960's, and there are a number of exact algorithms for product-form networks, as well as many approximation algorithms for non-product-form networks in the literature. In the first part of this section, we briefly review these exact methods, and then review the *mean value analysis* (MVA) method in detail. In the second part, we present some approximation algorithms for non-product-form CQNs.

2.6.1. Algorithms for product-form CQNs

As discussed in Section 2, product-form solutions are easily expressed and more effective than direct state space methods. However, it is still time consuming to compute the normalization constant by equation (2.10). Therefore, efficient algorithms have been developed to reduce the computation time. The MVA method is the most important algorithm among these efficient algorithms, and we will discuss this in detail

later. Besides MVA, there are many other popular algorithms for calculating performance measures of product-form CQNs existing in the literature and we briefly introduce two of them.

First, Buzen (1971) developed an efficient algorithm called the *convolution algorithm* to calculate the normalization constant $G(K)$. This algorithm iterates over nodes in the network and the number of possible customers at each node to compute $G(K)$.

The second algorithm for product-form CQNs is called the *RECAL algorithm*. RECAL algorithm denotes recursion by chain algorithm and was first introduced by Conway and Georganas (1986). An advantage of this algorithm is that it is suitable for networks with a large number of customer classes but a small number of nodes.

2.6.1.1. Mean value analysis. The MVA was developed by Reiser and Lavenberg (1980) for analysis of product-form CQNs. The most important advantage of this alogrithm is that it computes performance measures without using the normalization constant.

First, two important theorems are presented, which are the foundation for the MVA method. The first theorem is *Little's theorem* as shown below.

Theorem 5. *(Little's Theorem) The mean number of customers can be expressed by the throughput and the mean response time:*

$$\overline{L} = \lambda \overline{W}, \tag{2.40}$$

and the mean number of customers in a queue can be expressed by the throughput and the mean waiting time:

$$\overline{L}q = \lambda \overline{W}q. \tag{2.41}$$

This smart guess of the relation between two important performance measures was proved by Little (1961).

The second important theorem is the *arrival theorem* proved by Reiser and Lavenberg (1980).

Theorem 6. *(Arrival Theorem) For a single-class product-form CQN, the probability mass function (pmf) of the number of customers at ith node when there are K customers in the network equals to the pmf of the number of customers at ith node when there are $K-1$ customers in the network:*

$$Pr(\{k_1, \cdots, k_i, \cdots, k_M\}|K) = Pr(\{k_1, \cdots, k_i, \cdots, k_M\}|K-1), \tag{2.42}$$

For a multi-class product-form CQN, the relationship also holds for cth class of customers:

$$Pr_c(\{\vec{\mathbf{S}}_1, \cdots, \vec{\mathbf{S}}_i, \cdots, \vec{\mathbf{S}}_M\}|\vec{\mathbf{K}}) = Pr_c(\{\vec{\mathbf{S}}_1, \cdots, \vec{\mathbf{S}}_i, \cdots, \vec{\mathbf{S}}_M\}|\vec{\mathbf{K}} - \vec{e}_c), \tag{2.43}$$

where $\vec{e}_c$ is an $M \times 1$ vector that cth element is 1 and other elements are 0s.

Theorem 6 indicates an important property: At the moment a customer arrives at a node, this customer is not included in the network, which is same as the system with one fewer customer in the network. Hence, quantities of the network with K customers can be related to quantities of the network with $K-1$ customers. This is the base of the recursive procedure of MVA.

We introduce the basic MVA for single-class CQNs first, then this algorithm is extended to multi-class CQNs.

The basic equation derived from Theorems 5 and 6 is:

$$\overline{W}_i(K) = \frac{1}{\mu_i}\bar{L}_i(K) = \frac{1}{\mu_i}\overline{L}_i(K-1),\ i = 1, \cdots, M. \tag{2.44}$$

Equation (2.44) is valid for single-class CQN with FCFS nodes, LCFS nodes and PS nodes. For IS-nodes, $\overline{W}_i$ is:

$$\overline{W}_i(K) = \frac{1}{\mu_i}. \tag{2.45}$$

Currently, we only study FCFS case, and we may extend our study for other strategies in future.

Equation (2.44) needs to be revised for CQNs with parallel servers nodes. For FCFS discipline, $\overline{W}_i(K)$ can be expressed as:

$$\overline{W}_i(K) = \sum_{j=1}^{K} \frac{j}{\mu_i \varphi_i(j)} \pi_i(j-1|K-1),$$

$$\varphi_i = \begin{cases} j & j \leq m_i, \\ m_i & \text{otherwise.} \end{cases}$$

These equations can be inducted and rearranged as:

$$\begin{aligned} \overline{W}_i(K) &= \sum_{j=m_i}^{K} \frac{j\pi_i(j-1|K-1)}{\mu_i m_i} + \sum_{j=1}^{m_1-1} \pi_i(j-1|K-1) \\ &= \frac{1}{\mu_i m_i} \overline{L}_i(K) + \sum_{j=1}^{m_1-1} \pi_i(j-1|K-1) \\ &= \frac{1}{\mu_i m_i} \overline{L}_i(K-1) + \sum_{j=1}^{m_1-1} \pi_i(j-1|K-1) \\ &= \frac{1}{\mu_i m_i} \overline{L}_i(K-1) + \sum_{j=0}^{m_1-2} \pi_i(j|K-1). \end{aligned}$$

The expression of $\pi_i(j|K)$ is obtained in the lemma shown below:

Lemma 1. *(Baskett et al. (1975)) For $j = 1, \cdots, K$, the marginal probabilities satisfy:*

$$\begin{aligned} \pi_i(j|K) &= \frac{\lambda_i(K)}{\mu_i \varphi_i(j)} \pi_i(j-1|K-1), \\ \pi_i(0|K) &= \pi_i(0|0) - \sum_{j=1}^{K} \pi_i(j|K). \end{aligned} \tag{2.46}$$

Now, we can obtain the equation of $\overline{W}_i(K)$ as shown below:

$$\overline{W}_i(K) = \begin{cases} \frac{1}{\mu_i} \overline{L}_i(K-1) & m_i = 1, \\ \frac{1}{\mu_i m_i} \overline{L}_i(K-1) + \sum_{j=0}^{m_1-2} \pi_i(j|K-1) & m_i > 1. \end{cases} \tag{2.47}$$

The next lemma from Reiser and Lavenberg (1980) shows two additional equations to describe the MVA completely.

Lemma 2. *The overall throughput of the network with K customers in the network is:*

$$\lambda(K) = \frac{K}{\sum_{i=1}^{M} vr_i \overline{W}_i(K)}. \tag{2.48}$$

The mean number of customer at the ith node:

$$\overline{L}_i(K) = \lambda(K) \overline{W}_i(K) vr_i. \tag{2.49}$$

The complete MVA method for single-class product-form CQNs can be described in Algorithm 1.

Algorithm 1. Mean Value Analysis

STEP 1. Initialization

For $i = 1, \cdots M$ and $j = 1, \cdots, (m_i - 1)$, $\overline{L}_i(0) = 0$, $\pi_i(0|0) = 1$ and $\pi_i(j|0) = 0$.

STEP 2. Iteration for $k = 1, \cdots K$:

2.1 For $i = 1, \cdots M$, compute $\overline{W_i(k)}$ by using equation (2.47).

2.2 Compute the overall throughput $\lambda(k)$ by using equation (2.48) and $\pi_i(j|k)$ by equation (2.46).

2.3 For $i = 1, \cdots M$, compute the mean number of customers at ith node $\overline{L}_i(k)$ by equation (2.49).

Theorem 6 is also suitable for multi-class CQNs. We briefly present the MVA for multi-class CQNs here.

Algorithm 2. Mean Value Analysis for Multi-class CQN

STEP 1. Initialization

For $i = 1, \cdots, M$, $c = 1, \cdots, C$, $j = 1, \cdots, (m_i - 1)$, $\overline{L}_{ic}(\vec{\mathbf{0}}) = 0$, $\pi_i(0|\vec{\mathbf{0}}) = 1$, $\pi_i(j|\vec{\mathbf{0}}) = 0$.

STEP 2. Iteration for $\vec{\mathbf{k}} = \vec{\mathbf{0}}, \cdots, \vec{\mathbf{K}}$:

2.1 For $i = 1, \cdots M$ and $c = 1, \cdots, C$, compute the mean response time of cth class of customers at ith node:

$$\overline{T}_{ic}(\vec{\mathbf{k}}) = \begin{cases} \frac{1}{\mu_{ic}}[1 + \sum_{s=1}^{C} \overline{L}_{is}(\vec{\mathbf{k}} - \vec{e}_c)] & m_i = 1, \\ \frac{1}{\mu_{ic}}[1 + \sum_{s=1}^{C} \overline{L}_{is}(\vec{\mathbf{k}} - \vec{e}_c) + \sum_{j=0}^{m_i-2} (m_i - j - 1)\pi_i(j|\vec{\mathbf{k}} - \vec{e}_c)] & m_i > 1. \end{cases} \tag{2.50}$$

2.2 For $c = 1, \cdots, C$, compute the overall throughput per class:

$$\lambda_c(\vec{\mathbf{k}}) = \frac{L_c}{\sum_{i=1}^{M} vr_{ic}\overline{W}_{ic}(k)}. \tag{2.51}$$

2.3 For $i = 1, \cdots, M$, $c = 1, \cdots, C$, compute the mean number of cth class of customers at ith node:

$$\overline{L}_{ic}(\vec{\mathbf{k}}) = \lambda_c(\vec{\mathbf{k}})\overline{W}_{ic}(\vec{\mathbf{k}})vr_{ic}. \tag{2.52}$$

2.6.1.2. MVA extension. MVA has been developed and modified to solve CQNs under various assumptions. We briefly review several extensions of MVA.

First, Zahorjan and Wong (1981) extended MVA to the analysis of mixed product-form queueing networks with single server nodes. In this paper, the arrival theorem 6 has been extended to product-form OQNs. Then the MVA algorithm for CQNs can be modified to evaluate mixed queueing networks. Krzesinski et al. (1981) and Bruell et al. (1984) extended this result to multi-server nodes case.

Secondly, the MVA can be extended to evaluate product-form CQNs with load-dependent service times, which was first introduced by Reiser (1981). This extension straightforwardly replaces the load-independent service rate μ_i at ith node by the load-dependent service rate $\mu_i(j)$ when there are j customers at ith node. We briefly recap the MVA for single-class CQNs with load-dependent service, and the multi-class case can be simply extended from it.

Algorithm 3. Mean Value Analysis for Single-class CQNs with Load-dependent Service

STEP 1. Initialization

For $i = 1, \cdots M$, $\overline{L}_i(0) = 0$, $\pi_i(0|0) = 1$.

STEP 2. Iteration for $k = 1, \cdots K$*:*

2.1 For $i = 1, \cdots M$, *compute* $\overline{W_i(k)}$ *by:*

$$\overline{W}_i(k) = \sum_{j=1}^{k} \frac{1}{\mu_i(j)} \pi_i(i-1|k-1). \tag{2.53}$$

2.2 Compute the overall throughput $\lambda(k)$ *by equation (2.48) and* $\pi_i(j|k)$ *by:*

$$\begin{aligned} \pi_i(j|k) &= \frac{\lambda_i(k)}{\mu_i(j)\varphi_i(j)} \pi_i(j-1|k-1), \\ \pi_i(0|k) &= \pi_i(0|0) - \sum_{j=1}^{K} \pi_i(j|k). \end{aligned} \tag{2.54}$$

2.3 For $i = 1, \cdots M$, compute the mean number of customers at ith node $\overline{L}_i(k)$ by equation (2.49).

The third extension of MVA was developed for CQNs with a large number of customer classes. In these CQNs, exact methods we reviewed are time-consuming because the computation time of these methods grows exponentially with the number of classes. There are many approximation methods in the literature, and we briefly review the *Bard-Schweitzer approximation method* based on MVA. This method requires much less memory than exact methods and still gives accurate results. Bard (1979) and Schweitzer (1984) developed this approximation method based on MVA for multi-class CQNs with single server nodes. The basic idea can be described in the following three steps:

STEP 1. Start with an initial value $\overline{L}_{ic}(\vec{\mathbf{K}})$ for the given population vector $\vec{\mathbf{K}}$, and estimate $\overline{L}_{ic}(\vec{\mathbf{K}} - \vec{e}_s)$ for all classes;

STEP 2. Use equation (2.50) for single server nodes to estimate $\overline{L}_{ic}(\vec{\mathbf{K}})$, equation (2.51) to estimate $\lambda_c(\vec{\mathbf{K}})$ and equation (2.52) for a new value of $\overline{L}_{ic}(\vec{\mathbf{K}})$. This step is actually the one-step iteration of MVA for multi-class CQNs with given population vector $\vec{\mathbf{K}}$;

STEP 3. If the difference between the values of $\overline{L}_{ic}(\vec{\mathbf{K}})$s obtained in two interactions is less than the given error criterion, stop. Otherwise, go back to STEP 1.

Neuse and Chandy (1981) developed a method called *self-correcting approximation technique* (SCAT) as an extension of Bard-Schweitzer method. Zahorjan et al. (1988) proved that the SCAT is better than the Bard-Schweitzer method.

2.6.2. Algorithms for non-product-form CQNs

In real applications, many CQNs do not satisfy the local balance property (Property 1). For example, the distribution of inter-arrival and service times is usually

not exponentially distributed. Traditionally, methods based on discrete event simulation are used to estimate performance measures of these CQNs. However, simulation methods are usually time consuming and costly. In this section, we review some approximate methods for non-product-form CQNs as efficient alternatives of the simulation method.

There are many different types of non-product-form CQNs, and here we only present algorithms for non-product-form CQNs with non-exponentially distributed service times, which is related to our research.

The straightforward way to solve non-product-form CQNs with $\sim /G/1$ and $\sim /G/m\ FCFS$ nodes is to replace these nodes with $\sim /M/1$ and $\sim /M/m\ FCFS$ nodes and then apply methods of product-form CQNs. This is the basic idea of the *robustness method*. This method is based on an important *robustness property* that assumes changing of major parameters only leads minor changes in performance measures in CQNs with non-exponentially distributed service times. Suri (1983), Denning and Buzen (1978) did some studies about the robustness property of CQNs. Bolch et al. (1998) investigated more than 100 different non-product-form CQNs with $\sim /G/1$ and $\sim /G/m\ FCFS$ nodes. For these CQNs, they got small differences between non-product-form and product-form in most cases.

The second method for non-product-form CQNs containing single or multiple generally distributed service times nodes is called *Marie's method*, which is developed by Marie (1980). The main idea of this method is described below:

STEP 1. Consider a product-form CQN simply replacing $\sim /G/m\ FCFS$ nodes by $\sim /M/m\ FCFS$ nodes. The load-dependent arrival rates $\lambda_i(k)$s can be determined by an iterative procedure;

STEP 2. Consider a non-product-form CQN with $\lambda_i(k)/G/1\ FCFS$ nodes. A *Coxian distribution* is used to replace the general distribution. Load dependent

throughput and other system performance measures can be obtained by an iterative procedure. These results approximately estimate performance measures of the original CQN.

2.7. SOQN methodology review

As mentioned in Section 2, SOQNs are a set of queueing networks with additional resources. Each entering customer needs to be paired with this resource before beginning service. If no resources are available, customers wait in the external queue until a resource becomes available. This set of queueing networks can also be viewed as OQNs with a population constraint (Dallery (1990)). Obviously, SOQNs do not fulfill Property 1. Hence, there is no product-form solution. The approximate analysis for single-class SOQNs can be divided into three categories: *aggregation based methods*, *CQN based methods* and *matrix geometric based methods*. We review these three kinds of approximation methods respectively in the next subsection. On the other hand, the multi-class SOQN is a more complicated queueing network and there are not many mature methodologies for this in the literature. We briefly review some attempts to solve multi-class SOQNs.

2.7.1. Single-class SOQN

The first category of approximate analysis for single-class SOQNs is the aggregation based method. This method firstly introduced by Avi-Itzhak and Heyman (1973) to solve single-class SOQNs with exponentially distributed service times. A multiprogramming computer system was modeled as a CQN before this paper was published. The number of jobs in the system is determined by the capacity of the *central processing unit* (CPU). However, the number of jobs in the system is not fixed and the maximum number is determined by the capacity of CPU. Thus, Avi-Itzhak and

Heyman (1973) modeled the computer system as an OQN with an external queue. The first step of this aggregation method is to develop a CQN for this queueing network by adjusting the routing probabilities. Then, this CQN is replaced by a single load-dependent exponential server. The throughput of this server is obtained by calculating the throughput of the CQN for all possible numbers of jobs. Finally, the entire queueing network is solved as an $M/M(n)/N$ queue to obtain the queue length and waiting time of the external queue.

This approach was applied for modeling job shops with limited storage space by Buzacott (1974). Buzacott and Shanthikumar (1980) extended this aggregation method to analyze flexible manufacturing systems. Yao and Buzacott (1985) extended this approach to analyze flexible manufacturing systems with state-dependent routings. For SOQNs with generally distributed servers, the procedure is similar except that the load-dependent throughput is obtained by an *approximate mean value analysis* (AMVA) algorithm (Buitenhek et al. (2000)).

For single-class SOQNs with general servers, the CQN based method has been developed to estimate performance measures. This method was first proposed by Dallery (1990). The main idea of this method can be described as follows:

STEP 1: Approximate generally distributed service times by Coxian distributions, and assume the arrival process is Poisson;

STEP 2: Construct an equivalent CQN by treating additional resources as customers and adding a virtual service station or a synchronization station where customers are paired with resources;

STEP 3: This CQN is analyzed by Marie's method discussed in Section 2.

Different from the aggregation based method, the external queue is not estimated directly from the queue in front of the load-dependent server. The number of customers waiting in the external queue equals the number of customers denoted by a

birth-death process minus the number of resources in front of the synchronization station. The *birth rate* of this birth-death process equals the arrival rate of customers, and the *death rate* is equal to the state-dependent arrival rate of the resources from Marie's method.

The third category of approximate analysis for SOQNs is the matrix geometric based method.

For a Markov process with infinite number of states, exact solutions can only be obtained if this process has a repetitive structure for the global balance equations. This repetition is the foundation of a recursive solution for the steady state probabilities since state $i+1$ can be determined if state i is known in this repetitive structure. The repetitive structure is called *geometric* form. A Markov process with a geometric form structure is shown in a simple example.

Example 2. *Consider an $M/M/1$ queue with arrival rate λ, service rate μ. The Markov process of this queue is shown in figure 2.3.*

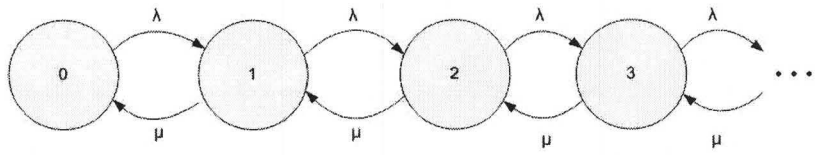

Figure 2.3. The Markov process of an $M/M/1$ queue

The global balance equations of this Markov process is:

$$\lambda\pi(0) = \mu\pi(1),$$

$$(\lambda + \mu)\pi(i) = \lambda\pi(i-1) + \mu\pi(i+1) \text{ where } i \geq 1.$$

The repetitive form is

$$\pi(i) = (\frac{\lambda}{\mu})^i\pi(0).$$

Because $\sum_{i=0}^{\infty} \pi(i) = 1$,

$$\pi(0) = \frac{1}{\sum_{i=0}^{\infty} (\frac{\lambda}{\mu})^i} = \frac{1}{1 + \frac{\lambda/\mu}{1-\lambda/\mu}} = 1 - \frac{\lambda}{\mu}.$$

It is easy to obtain the well-known performance measures as follows:

$$L = \sum_{i=1}^{\infty} i\pi(i) = \sum_{i=1}^{\infty} i(1 - \frac{\lambda}{\mu})(\frac{\lambda}{\mu})^i = \frac{\lambda}{\mu - \lambda},$$
$$W = \frac{L}{\lambda} = \frac{1}{\lambda - \mu}.$$

Neuts (1981) first proposed a standard solution procedure to describe and analyze $GI/M/1$ and $M/G/1$ queues that have a similar repetitive structure, which is called *matrix geometric method* (MGM). This method is part of the foundation of our research and will be discussed in detail later. Ramaswami (1988) provided a numerically stable approach based on MGM to the computation of steady-state probabilities of an $M/G/1$ queue. Akar et al. (1998) extended this method to a unifying generalized state-space approach for different types of $M/G/1$ queues.

The MGM has been applied to solve SOQNs by Buitenhek (1998). Since the MGM can only solve the single-class, two-station networks practically, an approximate method was developed to solve SOQNs with more than two stations. At first, the network is divided into two subnetworks. Then, the MVA algorithm is used to determine the load-dependent throughput rates for each network. Finally, the MGM is used to solve the two-station SOQN with load-dependent servers. Jia and Heragu (2009) developed an exact method based on MGM to solve the single-class SOQN for some simple cases. For the multi-class SOQN, all of the classes are aggregated into a new virtual class and all arrival and service process parameters are determined. Then, the MGM is applied to solve this new single-class SOQN. Comparing with Buitenhek

(1998)'s method and Dallery (1990)'s method, the method by Jia and Heragu (2009) performs better for several examples considered in their paper.

2.7.2. Multi-class SOQN

There are two main directions in the study of multi-class SOQNs.

The first direction is the multi-class SOQNs with dedicated resources, which means each class of customers has a designated type of resource. Lazowska and Zahorjan (1982) and Brandwajn (1982) first proposed an approximation approach for this type of multi-class SOQNs. This approach analyzes a multi-class SOQN (in which every class of customers has an independent population constraint) by evaluating one customer class at a time. The impact on other classes is presented by the average load. Thomasian and Bay (1984) presented another method. In this method, different classes of customers are still studied one by one. The difference between the approach by Lazowska and Zahorjan (1982) and Brandwajn (1982) is that the impact on other classes is based on the utilization of other classes visiting the station. This approach is also not efficient if there is only one general type of resource in the system.

As mentioned, a single-class SOQN can be solved by an approximate analysis based on Marie's method. If Marie's method can be extended to multi-class CQNs, the multi-class SOQNs with a general resource can be solved. Perros et al. (1992) extended this method to multi-class SOQNs by assuming the corresponding OQN is a BCMP-type network due to the computational difficulty of the general case. In order to solve general mutli-class SOQNs, Buitenhek et al. (2000) proposed an approximation approach to transform the multi-class SOQN into a multi-class CQN by assuming the resource changes class at the synchronization station. A modified AMVA algorithm has been developed to analyze the performance of this CQN. However, the method is not efficient when the number of classes is large because the

modified AMVA requires more iteration steps depending on the number of customer classes.

Baynat and Dallery (1996) proposed another extension of Marie's method for multi-class CQNs with generally distributed service times, state-dependent routing, and *fork-join* mechanisms. Jia and Heragu (2009) applied this approach in their solution for multi-class SOQNs with dedicated resources.

The second direction is the multi-class SOQNs with a general resource, which means all classes of customers share a common type of resource. Buitenhek (1998) and Buitenhek et al. (2000) presented an important work on this type of multi-class SOQNs. The most important contribution of these two papers is that they proved multi-class SOQNs with a general resource are unnecessarily analyzed in the same complicated way as multi-class SOQNs with dedicated resources before. Therefore, two simplified approaches are proposed in these papers. The first approach is the partial reduction approach, where the multi-class SOQN is aggregated into a two-class SOQN. The second approach is the complete reduction approach, where the multi-class SOQN is aggregated into a single-class SOQN. Jia and Heragu (2009) applied the second approach to solve multi-class SOQNs. They utilized the MGM approach for solving single-class SOQNs to solve multi-class SOQNs with general resources and effectively solved networks with generally distributed inter-arrival and service times.

CHAPTER 3

PERFORMANCE EVALUATION OF AS/RS AND AVS/RS WITH TIER-CAPTIVE VEHICLES

3.1. Introduction

In this chapter, we apply existing models for the analysis of AS/RS and AVS/RS with tier-captive vehicles. The objective of applying analytical models for the performance analysis of the two systems is twofold.

First, we demonstrate the power and application of a previously developed analytical model for *warehouse concepting* – exploring alternative configurations, estimating the performance of each for an assumed demand distribution, and performing sensitivity analysis. Concepting has been traditionally done using simulation models and as a result, designers have been able to evaluate only a handful number of configurations before settling on a final design.

Second, we demonstrate the use of an OQN model to answer key design questions. The key design questions we seek to answer include questions such as:

- Is the AVS/RS or AS/RS better for a given scenario?
- For a given warehouse application, how should the reserve area be configured? How many aisles, columns and levels are required?
- How many autonomous devices (cranes, lifts and vehicles) are required to meet the requirements of throughput capacity, cycle times, and S/R device utilizations?

- Should the high-bay area be an integrated entity, or should it be divided into zones (based on aisles, columns or tiers)? If it is the latter, how should the automated devices be allocated to the different zones?

In this chapter, we use analytical models to answer the first four design questions.

3.2. A comparison of AS/RS and AVS/RS

The different load movement patterns make the AVS/RS more flexible than an AS/RS. In a typical AS/RS, an aisle-captive crane moves the unit loads into and out of a storage space in the corresponding aisle. Because we need one crane for each aisle, this could lead to higher capital costs and lower utilization of material handling devices. Unlike storage cranes in the AS/RS, AVS/RS vehicles can access any aisle in any tier in the tier-to-tier configuration (more on the two possible AVS/RS configurations later) and any aisle in a designated tier in the tier-captive configuration. Because additional autonomous vehicles and lifts can be added or removed as desired in the AVS/RS and vehicles are not assigned to any specific aisle, a potential advantage of an AVS/RS compared to an AS/RS is the flexibility to satisfy the throughput requirement in different applications.

The second advantage of an AVS/RS is modularization. Because cranes complete all the S/R operations in AS/RS, it is hard to modify the system configuration. A small change in the AS/RS leads to a costly redesign of the entire system in many applications. On the other hand, the AVS/RS is highly modular. Different functional areas can be redesigned easily with minimal impact on other areas. For example, the number of lifts can be changed to satisfy higher throughput requirements while keeping other configuration parameters constant.

Another advantage of an AVS/RS relates to the dispatching transactions rules. In an AS/RS, retrieval transactions form individual queues in different storage aisles.

However, storage transactions may or may not be segregated by aisles depending on the storage policy and system configuration (Malmborg (2002)). In comparison, all buffered S/R requests are in a single queue in an AVS/RS. Pooling both storage and retrieval transactions in a single queue may enable AVS/R systems to achieve higher proportion of S/R cycles using dual commands (Malmborg and Altassan (1998)). This allows the AVS/RS to be easily expanded or contracted depending upon the throughput requirement. On the other hand, this feature may be a disadvantage for AVS/R systems because S/R transactions in an FCFS queue may use locations on different storage tiers, which means vehicles may take a longer time to complete the S/R transactions.

3.3. Application of analytical open queueing network model for analyzing AS/RS and AVS/RS with tier-captive vehicles

As we discussed in Chapter 1, queuing network models are powerful tools in estimating key performance measures of discrete-event, multistage service systems. Customers arriving from the outside world enter the system to complete several stages of service, and then leave the system. The main components of a queuing network are servers, queues and customers. Several types of queuing networks are available in the literature - OQN, CQN, and SOQN. The CQN has a population constraint and implicitly assumes there are infinite customers just outside the network. The population constraint is enforced by pairing each incoming customer with a limited number of another resource, such as a vehicle, which stays with the customer until service is completed at the last stage. When the customer leaves the system, the resource returns to the beginning of the network ready to be paired with a new customer. The number of resources is finite, thus enforcing the population constraint in a CQN. In an OQN, customers arriving from the outside receive service at multiple

stages of servers, and then depart from the system. The OQN implicitly assumes there is an infinite number of these additional resources so that an arriving customer never has to wait in the external queue. SOQN, on the other hand, models the more realistic scenario, where a customer may have to wait for the resource or vice-versa and provides better estimates of total cycle time and work-in-process (WIP) inventory (Jia and Heragu (2009)).

Application of the OQN to an AVS/RS with tier-captive vehicles is illustrated in figure 3.1. The server S1 represents the horizontal travel time of the autonomous vehicle on the ground floor between the load/unload point and the storage position. The server S2 represents the lift (vertical) travel time and S3 represents the horizontal travel time (again of the autonomous vehicle) on any tier other than the ground floor.

It is easy to model the AS/RS also using the OQN model, if we visualize the AS/RS as the AVS/RS with tier-captive vehicles tilted by 90 degrees with just one server (crane) in each tier. The aisles in the AS/RS become the tiers in the AVS/RS after the 90 degree tilt. Similarly, the server representing the lift in the AVS/RS now represents a conveyor in the AS/RS (see figure 3.2).

Many methods have been developed to analyze an OQN. Exact solution is only possible for networks with exponential inter-arrival and service time distributions. However, external arrival processes of complex queuing networks, including those seen in automated warehouses need not be Poisson and the service time distributions need not be exponential. To solve general queuing networks, several approximation methods are available in the literature. Among them, the parametric decomposition (PD) method is very popular and effective. It approximates non-renewal processes by stationary renewal processes. A renewal process means every time an event occurs, the process renews itself and starts all over again. The Poisson process is a special case of a renewal process where time between occurrences is exponentially distributed.

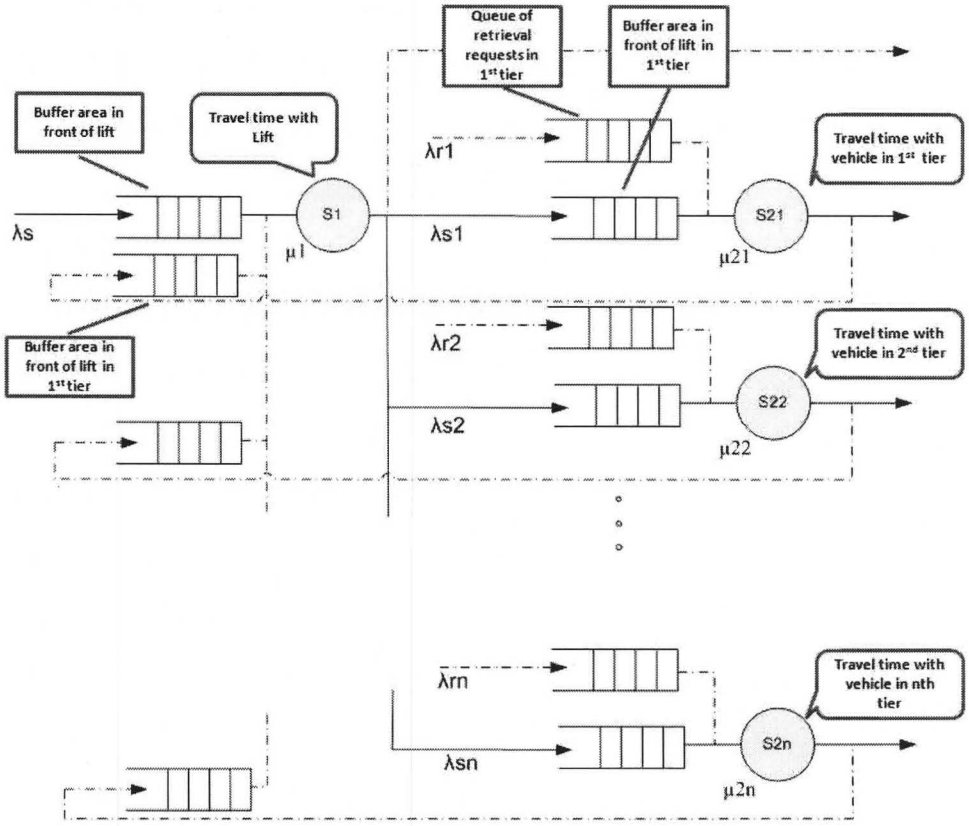

Figure 3.1. Modeling an AVS/RS as an OQN

Typically, a complex stochastic process has one or more embedded renewal processes, which allows the process to be decomposed into smaller independent systems. For example, Markovian networks can be decomposed into subsystems exactly, while general networks can only be decomposed approximately. By considering only the first two moments - mean value and squared coefficient of variation (SCV) - of all arrival and service processes, the PD method decomposes a complex queuing network into several isolated queues or subsystems (Kuehn (1979)).

QNA (Queuing Network Analyzer) is a software package to analyze different kinds of OQNs. The basic idea of QNA is the PD method for analyzing OQNs, which

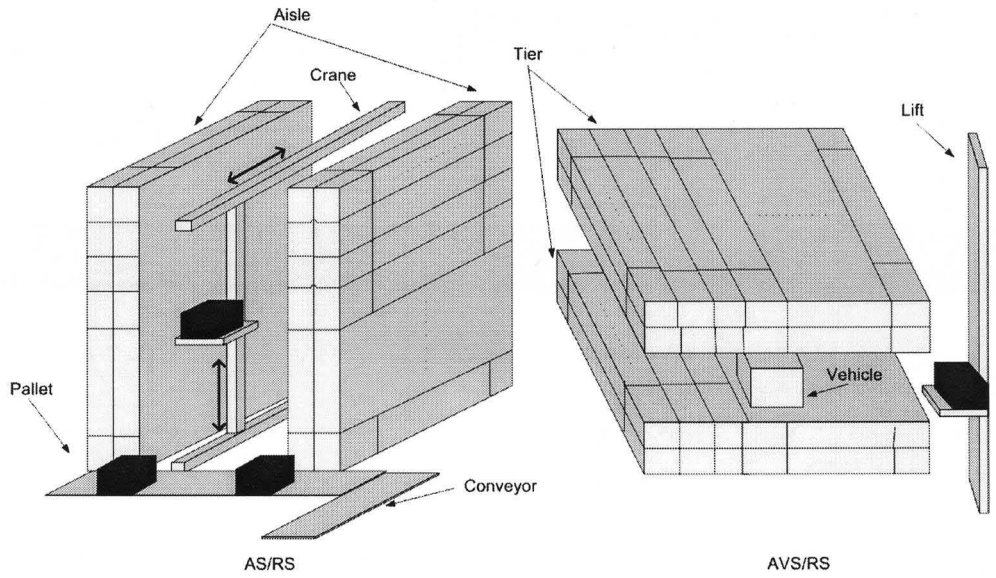

Figure 3.2. A 90 degree tilt of the AVS/RS with just one vehicle per tier transforms it into an AS/RS

was developed by Whitt (1983). It was originally intended for use in communication systems, but was later extended to model discrete parts manufacturing systems. The approximation method in QNA decomposes the queuing network into several stochastically independent $GI/G/m$ queues. A $GI/G/m$ queue indicates that the arrival process of the queue with m multiple servers is a renewal process with general distribution (GI), and the distribution of service time is also general (G). Once the first two moments of the inter-arrival time of each customer type and its routing are given, QNA calculates the first two moments of the *effective* inter-arrival times of customers at each node. QNA also calculates the first two moments of the effective service times at each node and then analyzes each node as an independent $GI/G/1$ or $GI/G/m$ queue to estimate performance measures of this node. Finally, performance measures of the entire queuing network are estimated by synthesizing performance measures of these independent $GI/G/1$ or $GI/G/m$ queues (Whitt (1983)).

MPA, an extension of QNA, and described in Meng and Heragu (2004), is an analytical model specifically designed to evaluate the performance of a manufacturing system. In addition to three key network operations captured in QNA, superposition, decomposition and departure, MPA captures another key operation - that of batching. It considers numerous real-world factors such as setup times, server failures, empty travel time of discrete material handling devices and provides reasonably accurate estimates of performance measures. The core of MPA is to solve equations of the first two moments of inter-arrival and service times for each node (Equations (29) - (30) in Meng and Heragu (2004)). We use MPA in this chapter to estimate performance measures of an AS/RS and an AVS/RS.

3.4. Use of analytical models for warehouse design conceptualization

In this section, we design a set of experiments using data from two companies to demonstrate how the analytical models presented in Section 3 can be used for design conceptualization. Design conceptualization is a key step in the design of a warehouse. Most designers begin with a particular technology they are familiar with (for example, AS/RS or AVS/RS), one or two configurations they have previously employed with this technology, and modify it slightly by scaling the design up or down to meet the throughput requirements of the current scenario. They then develop simulation models and verify/validate their design on their chosen configuration(s) to demonstrate that the customer throughput requirements can be met and select one of these designs depending upon the system integrator and customer preferences. While this approach provides one or two single usable designs, it may not necessarily be the best design because only a handful number of configurations are tested using a material handling system familiar to the system integrator. Use of MPA allows the integrator to test two alternate material handling technologies and numerous configurations -

many more than is possible via simulation - and estimate the performance measures and select the best one from among a larger set of candidate designs. We believe this approach allows the system integrator to arrive at a significantly better design than what they currently propose to their customers.

3.4.1. AVS/RS parameters

Using data from a warehouse implementation of an AVS/RS by Savoye Logistics, we show how to prepare or calculate the input data required by the analytical model presented in Section 3. The elements of the warehouse design are shown in tables 3.1 - 3.4. Figure 3.3 shows these elements on one tier of an AVS/RS.

Table 3.1. AVS/RS rack system data

Number of tiers (T)	7	
Number of aisles (A)	42	
Number of bays (B)	12	
Number of storage positions per bay (n_B)	3	
Distance between two bays (B_d)	$3030mm$	$9.94ft$
Distance between two aisles (D_a)	$4256mm$	$13.96ft$
Pallet height (H_p)	$2000mm$	$6.56ft$
Distance between level-pallet (Z_p)	$350mm$	$1.15ft$

Table 3.2. Autonomous vehicle data

Vehicle speed (V_v)	$2.5m/s$	$8.2ft/s(492ft/min)$
Vehicle acceleration/deceleration (a_v)	$0.5m/s^2$	$0.167ft/s^2$

Table 3.3. Lift data

Lift speed (V_l)	$1.5m/s$	$4.9ft/s(294ft/min)$
Lift acceleration/deceleration (a_l)	$0.5m/s^2$	$0.167ft/s^2$
Pallet transfer from buffer area to lift (t_2)	$10s$	

The effective vehicle travel times and the effective lift travel times are two parameters needed in the OQN model.

Table 3.4. Pallet data

Length (P_L)	$1200mm$	$3.94ft$
Width (P_W)	$800mm$	$2.62ft$
Height (P_H)	$2000mm$	$6.56ft$

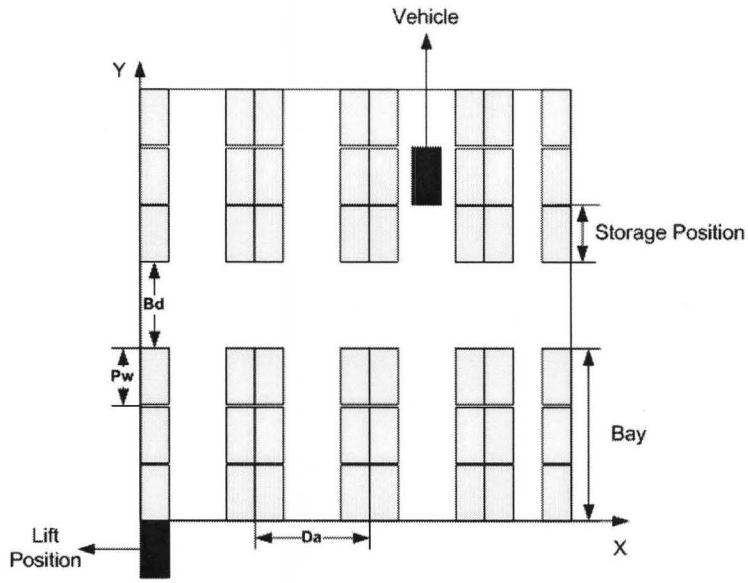

Figure 3.3. Elements on a tier of an AVS/RS

3.4.1.1. Estimating effective vehicle travel time. The vehicle travel time on a given tier is determined by three factors. The first factor is the command type. We assume that the vehicle can execute either a storage request (S) or a retrieval request (R). The second factor determines the position of the available vehicle. We denote this factor by a pair of coordinates ($x_v(i)$, $y_v(j)$), where i denotes ith aisle, and j denotes jth column. The number of columns C is equal to the number of storage positions per aisle and is given by:

$$C = B \times n_B = 51, \tag{3.1}$$

and the average width of each storage position uL is:

$$uL = \frac{(B_d - n_B P_w)(B-1)}{n_B B} + P_w = 1.002m. \tag{3.2}$$

The third factor is the destination/origin of S/R requests, which is denoted by a pair of coordinates $(x_p(i),\ y_p(j))$. For example, $(S,\ (x_v(1),\ y_v(10)),\ (x_p(2),\ y_p(20)))$ describes a storage request with an available vehicle at the first aisle and 10th column and the designated storage position at the second aisle and 20th column.

There are $2 \times A \times C \times A \times C$ or $9,176,328$ possible scenarios for this AVS/RS. Let Pr_{vk} and t_{vk} respectively denote the probability and vehicle travel time of kth scenario. According to *Baye's theorem*, the effective vehicle travel time is:

$$\mathbb{E}(t_v) = \sum_{k=1}^{2\times A^2 \times C^2} Pr_{vk} t_{vk}. \tag{3.3}$$

We assume all arrival processes are Poisson. The probability of a storage request at each tier Pr_s and the probability of a retrieval request at each tier Pr_r are:

$$\begin{aligned} Pr_s &= \frac{\lambda_s/T}{\lambda_s/T + \lambda_r/T} = \frac{\lambda_s}{\lambda_s + \lambda_r}, \\ Pr_r &= \frac{\lambda_r/T}{\lambda_s/T + \lambda_r/T} = \frac{\lambda_r}{\lambda_s + \lambda_r}, \end{aligned} \tag{3.4}$$

where λ_s and λ_r are overall S/R throughput. Additionally, note that the S/R policy is purely random, and therefore the probability that an available vehicle is at any S/R position is the same. Hence, the Pr_{vk} is obtained as:

$$Pr_{vk} = \begin{cases} Pr_s \frac{1}{A^2}\frac{1}{C^2} = \frac{\lambda_s}{\lambda_s+\lambda_r}\frac{1}{A^2}\frac{1}{C^2} & \text{storage request,} \\ Pr_r \frac{1}{A^2}\frac{1}{C^2} = \frac{\lambda_r}{\lambda_s+\lambda_r}\frac{1}{A^2}\frac{1}{C^2} & \text{retrieval request.} \end{cases} \tag{3.5}$$

On the other hand, t_{vk} is determined by the available vehicle position and the destination/origin of S/R requests. We use P_v and P_p to denote the available vehicle

position and the designated position for kth scenario respectively. Figure 3.4 shows the typical travel paths for S/R transactions.

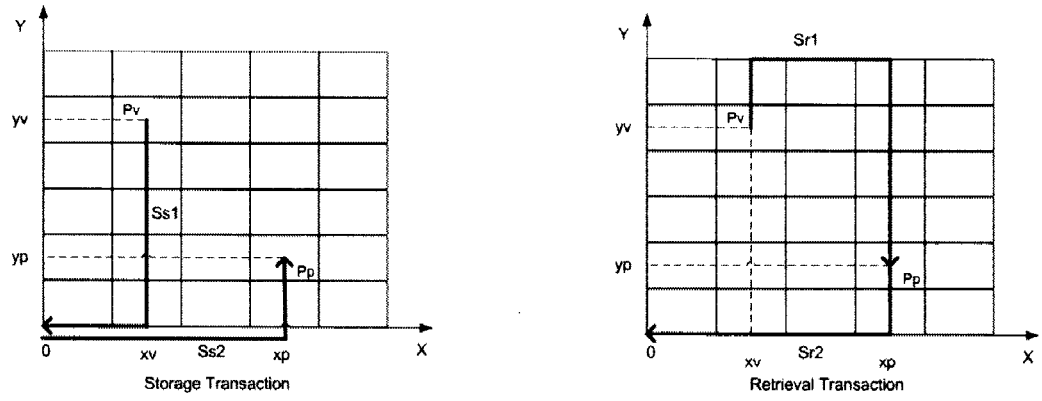

Figure 3.4. Travel path in a tier for S/R requests

Here we simplify the layout of a tier to a rectangle shape with grids. Each grid denotes a possible storage position with length as D_a and width as uL. For example, if the vehicle is in the ith aisle, jth column, the position of this vehicle in figure 3.4 is in the grid of ith row and jth column, and

$$\begin{aligned} x_v(i_v) &= i_v D_a, \ i_v = 0, \cdots, A-1, \\ y_v(j_v) &= j_v uL, \ j_v = 0, \cdots, C-1. \end{aligned} \tag{3.6}$$

For a designated position Pp, $(x_p(i_p), y_p(j_p))$ can also be obtained by equation (3.6).

Before estimating the t_{vk}s, we develop some formulae of travel times between different positions. The first formula is the travel time on the X-axis between any two aisles t_x. By considering the acceleration/deceleration of vehicles, it is known that when the distance between two aisles s_x is shorter than s_a ($s_a = t_a V_v = 2.5 t_a = 12.5m$, $t_a = \frac{V_v}{a_v} = 5s$):

$$t_x = 2\sqrt{\frac{s_x}{a_v}}. \tag{3.7}$$

Otherwise,

$$t_x = \frac{s_x - s_a}{V_v} + 2t_a. \tag{3.8}$$

This vehicle speed is shown in Figure 3.5.

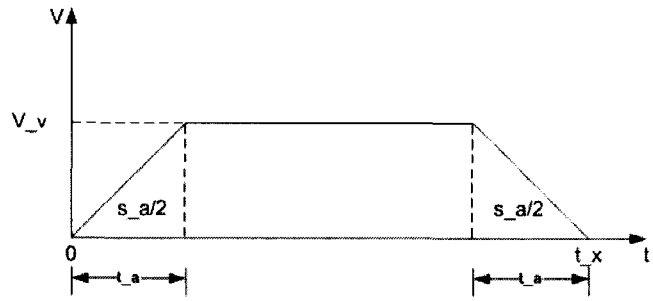

Figure 3.5. Vehicle speed

The second formula is the travel time t_y on Y-axis between any two columns s_y. Similar to equations (3.7) and (3.8):

$$t_y = \begin{cases} 2\sqrt{\frac{s_y}{a_v}} & s_y < s_a, \\ \frac{s_y - s_a}{V_v} + 2t_a & \text{otherwise.} \end{cases} \tag{3.9}$$

As shown in figure 3.4, the storage travel path contains two parts, s_{s1} and s_{s2}. s_{s1} denotes the vehicle travel path from the position where it completed the transaction to the lift position. We assume that all the lifts are located next to each other in the left-front corner of the reserve storage area, or at the origin. s_{s2} denotes the path for the vehicle to transfer the pallet from the lift position to the designated storage position.

$$\begin{aligned} s_{s1}(i_v, j_v, i_p, j_p) &= y_v(j_v) + x_v(i_v) = j_v uL + i_v D_a, \\ s_{s2}(i_v, j_v, i_p, j_p) &= x_p(i_p) + y_p(j_p) = i_p D_a + j_p uL. \end{aligned} \tag{3.10}$$

The corresponding travel times t_{s1} and t_{s2} can be obtained by equations (3.7) - (3.9) respectively:

$$t_{s1}(i_v, j_v, i_p, j_p) = \begin{cases} 2\sqrt{\frac{j_v uL}{a_v}} + 2\sqrt{\frac{i_v D_a}{a_v}} & j_v uL < s_a, i_v D_a < s_a, \\ 2\sqrt{\frac{j_v uL}{a_v}} + \frac{i_v D_a - s_a}{V_v} + 2t_a & j_v uL < s_a, i_v D_a \geq s_a, \\ \frac{j_v uL - s_a}{V_v} + 2t_a + 2\sqrt{\frac{i_v D_a}{a_v}} & j_v uL \geq s_a, i_v D_a < s_a, \\ \frac{j_v uL - s_a}{V_v} + 2t_a + \frac{i_v D_a - s_a}{V_v} + 2t_a & j_v uL \geq s_a, i_v D_a \geq s_a, \end{cases} \tag{3.11}$$

$$t_{s2}(i_v, j_v, i_p, j_p) = \begin{cases} 2\sqrt{\frac{i_p D_a}{a_v}} + 2\sqrt{\frac{j_p uL}{a_v}} & i_p D_a < s_a, j_p uL < s_a, \\ 2\sqrt{\frac{i_p D_a}{a_v}} + \frac{j_p uL - s_a}{V_v} + 2t_a & i_p D_a < s_a, j_p uL \geq s_a, \\ \frac{i_p D_a - s_a}{V_v} + 2t_a + 2\sqrt{\frac{j_p uL}{a_v}} & i_p D_a \geq s_a, j_p uL < s_a, \\ \frac{i_p D_a - s_a}{V_v} + 2t_a + \frac{j_p uL - s_a}{V_v} + 2t_a & i_p D_a \geq s_a, j_p uL \geq s_a. \end{cases} \tag{3.12}$$

The vehicle travel time for a storage transaction $t_v(S, i_v, j_v, i_p, j_p) = t_{s1}(i_v, j_v, i_p, j_p) + t_{s2}(i_v, j_v, i_p, j_p)$ has 16 possible combinations.

The retrieval transaction path (see the right side of figure 3.4) also includes two parts: s_{r1} and s_{r2}. s_{r1} is the path for the vehicle to travel from where it completed the last transaction to the retrieval request position. There are three parts in s_{r1} when the vehicle and the origin of the retrieval request are not in the same aisle: travel distance to the X-axis, travel distance on the X-axis and travel distance on the X-axis to the retrieval position. Obviously, S_{r1} is the shorter part of a rectangular movement path. If

$$y_v(j_v)+y_p(j_p)+|x_v(i_v)-x_p(i_p)| \geq ((C-1)uL-y_v(j_v))+((C-1)uL-y_p(j_p))+|x_v(i_v)-x_p(i_p)|,$$

or $j_v + j_p \geq C - 1$, the vehicle travels from its last position to the X-axis and then travels to the designated storage position. Otherwise, the vehicle travels to the opposite axis and then travels to the designated storage position. s_{r2} denotes the path for the vehicle to transfer the pallet from the designated retrieval position to the lift position.

$$s_{r1}(i_v, j_v, i_p, j_p) = \begin{cases} (j_v + j_p)uL + |i_v - i_p|D_a & j_v + j_p \geq C - 1, i_v \neq i_p, \\ (2C - 2 - j_v - j_p)uL + |i_v - i_p|D_a & j_v + j_p < C - 1, i_v \neq i_p, \\ |y_v(j_v) - y_p(j_p)| = |j_v - j_p|uL & i_v = i_p, \end{cases}$$

$$s_{r2}(i_v, j_v, i_p, j_p) = y_p(j_p) + x_p(i_p) = j_p uL + i_p D_a. \tag{3.13}$$

The corresponding travel times t_{r1} and t_{r2} can be obtained by equation (3.7) - (3.9) respectively. Table 3.5 shows t_{r1} under different conditions.

$$t_{r2}(i_v, j_v, i_p, j_p) = \begin{cases} 2\sqrt{\frac{i_p D_a}{a_v}} + 2\sqrt{\frac{j_p uL}{a_v}} & i_p D_a < s_a, j_p uL < s_a, \\ 2\sqrt{\frac{i_p D_a}{a_v}} + \frac{j_p uL - s_a}{V_v} + 2t_a & i_p D_a < s_a, j_p uL \geq s_a, \\ \frac{i_p D_a - s_a}{V_v} + 2t_a + 2\sqrt{\frac{j_p uL}{a_v}} & i_p D_a \geq s_a, j_p uL < s_a, \\ \frac{i_p D_a - s_a}{V_v} + 2t_a + \frac{j_p uL - s_a}{V_v} + 2t_a & i_p D_a \geq s_a, j_p uL \geq s_a. \end{cases} \tag{3.14}$$

Hence, the effecitve vehicle travel time for a retrieval transaction $t_v(R, i_v, j_v, i_p, j_p) = t_{r1}(i_v, j_v, i_p, j_p) + t_{r2}(i_v, j_v, i_p, j_p)$ has 72 combinations.

3.4.1.2. Estimating effective lift travel time. The lift travel time between any two tiers is also determined by three factors: S/R request type (S or R), current lift position ($y_l(m)$) and destination/origin tier ($y_s(m)/y_r(m)$), where m denotes mth tier. For instance, (S, $y_l(2)$, $y_s(4)$) means the available lift is on the second tier currently, and the storage destination is the 4th tier.

Table 3.5. t_{r1} under different conditions

$t_{r1}(i_v, j_v, i_p, j_p)$	Condition
	$j_v + j_p \geq C+1,\ i_v \neq i_p$
$2\sqrt{\frac{j_v uL}{a_v}} + 2\sqrt{\frac{j_p uL}{a_v}} + 2\sqrt{\frac{\lvert i_v - i_p\rvert D_a}{a_v}}$	$j_v uL < s_a, j_p uL < s_a,\ \lvert i_v - i_p\rvert D_a < s_a$
$2\sqrt{\frac{j_v uL}{a_v}} + \frac{j_p uL - s_a}{V_v} + 2t_a + 2\sqrt{\frac{\lvert i_v - i_p\rvert D_a}{a_v}}$	$j_v uL < s_a, j_p uL \geq s_a,\ \lvert i_v - i_p\rvert D_a < s_a$
$\frac{j_v uL - s_a}{V_v} + 2t_a + 2\sqrt{\frac{j_p uL}{a_v}} + 2\sqrt{\frac{\lvert i_v - i_p\rvert D_a}{a_v}}$	$j_v uL \geq s_a, j_p uL < s_a,\ \lvert i_v - i_p\rvert D_a < s_a$
$\frac{j_v uL - s_a}{V_v} + 2t_a + \frac{j_p uL - s_a}{V_v} + 2t_a + 2\sqrt{\frac{\lvert i_v - i_p\rvert D_a}{a_v}}$	$j_v uL \geq s_a, j_p uL \geq s_a,\ \lvert i_v - i_p\rvert D_a < s_a$
$2\sqrt{\frac{j_v uL}{a_v}} + 2\sqrt{\frac{j_p uL}{a_v}} + \frac{\lvert i_v - i_p\rvert D_a - s_a}{V_v} + 2t_a$	$j_v uL < s_a, j_p uL < s_a,\ \lvert i_v - i_p\rvert D_a \geq s_a$
$2\sqrt{\frac{j_v uL}{a_v}} + \frac{j_p uL - s_a}{V_v} + 2t_a + \frac{\lvert i_v - i_p\rvert D_a - s_a}{V_v} + 2t_a$	$j_v uL < s_a, j_p uL \geq s_a,\ \lvert i_v - i_p\rvert D_a \geq s_a$
$\frac{j_v uL - s_a}{V_v} + 2t_a + 2\sqrt{\frac{j_p uL}{a_v}} + \frac{\lvert i_v - i_p\rvert D_a - s_a}{V_v} + 2t_a$	$j_v uL \geq s_a, j_p uL < s_a,\ \lvert i_v - i_p\rvert D_a \geq s_a$
$\frac{j_v uL - s_a}{V_v} + 2t_a + \frac{j_p uL - s_a}{V_v} + 2t_a + \frac{\lvert i_v - i_p\rvert D_a - s_a}{V_v} + 2t_a$	$j_v uL \geq s_a, j_p uL \geq s_a,\ \lvert i_v - i_p\rvert D_a \geq s_a$
	$j_v + j_p < C+1,\ i_v \neq i_p$
$2\sqrt{\frac{(C-1-j_v)uL}{a_v}} + 2\sqrt{\frac{(C-1-j_p)uL}{a_v}} + 2\sqrt{\frac{\lvert i_v - i_p\rvert D_a}{a_v}}$	$(C-1-j_v)uL < s_a,\ (C-1-j_p)uL < s_a,\ \lvert i_v - i_p\rvert D_a < s_a$
$2\sqrt{\frac{(C-1-j_v)uL}{a_v}} + \frac{(C-1-j_p)uL - s_a}{V_v} + 2t_a + 2\sqrt{\frac{\lvert i_v - i_p\rvert D_a}{a_v}}$	$(C-1-j_v)uL < s_a,\ (C-1-j_p)uL \geq s_a,\ \lvert i_v - i_p\rvert D_a < s_a$
$\frac{(C-1-j_v)uL - s_a}{V_v} + 2t_a + 2\sqrt{\frac{(C-1-j_p)uL}{a_v}} + 2\sqrt{\frac{\lvert i_v - i_p\rvert D_a}{a_v}}$	$(C-1-j_v)uL \geq s_a,\ (C-1-j_p)uL < s_a,\ \lvert i_v - i_p\rvert D_a < s_a$
$\frac{(C-1-j_v)uL - s_a}{V_v} + 2t_a + \frac{(C-1-j_p)uL - s_a}{V_v} + 2t_a + 2\sqrt{\frac{\lvert i_v - i_p\rvert D_a}{a_v}}$	$(C-1-j_v)uL \geq s_a,\ (C-1-j_p)uL \geq s_a,\ \lvert i_v - i_p\rvert D_a < s_a$
$2\sqrt{\frac{(C-1-j_v)uL}{a_v}} + 2\sqrt{\frac{(C-1-j_p)uL}{a_v}} + \frac{\lvert i_v - i_p\rvert D_a - s_a}{V_v} + 2t_a$	$(C-1-j_v)uL < s_a,\ (C-1-j_p)uL < s_a,\ \lvert i_v - i_p\rvert D_a \geq s_a$
$2\sqrt{\frac{(C-1-j_v)uL}{a_v}} + \frac{(C-1-j_p)uL - s_a}{V_v} + 2t_a + \frac{\lvert i_v - i_p\rvert D_a - s_a}{V_v} + 2t_a$	$(C-1-j_v)uL < s_a,\ (C-1-j_p)uL \geq s_a,\ \lvert i_v - i_p\rvert D_a \geq s_a$
$\frac{(C-1-j_v)uL - s_a}{V_v} + 2t_a + 2\sqrt{\frac{(C-1-j_p)uL}{a_v}} + \frac{\lvert i_v - i_p\rvert D_a - s_a}{V_v} + 2t_a$	$(C-1-j_v)uL \geq s_a,\ (C-1-j_p)uL < s_a,\ \lvert i_v - i_p\rvert D_a \geq s_a$
$\frac{(C-1-j_v)uL - s_a}{V_v} + 2t_a + \frac{(C-1-j_p)uL - s_a}{V_v} + 2t_a + \frac{\lvert i_v - i_p\rvert D_a - s_a}{V_v} + 2t_a$	$(C-1-j_v)uL \geq s_a,\ (C-1-j_p)uL \geq s_a,\ \lvert i_v - i_p\rvert D_a \geq s_a$
	$i_v = i_p$
$2\sqrt{\frac{\lvert j_v - j_p\rvert uL}{a_v}}$	$\lvert j_v - j_p\rvert uL < s_a$
$\frac{\lvert j_v - j_p\rvert uL - s_a}{V_v} + 2t_a$	$\lvert j_v - j_p\rvert uL \geq s_a$

There are $2 \times T \times (T-1)$ or 84 possible scenarios for the AVS/RS we consider. The effective lift travel time $\mathbb{E}(t_l)$ can be obtained by:

$$\mathbb{E}(t_l) = \sum_{k=1}^{2\times T\times(T-1)} Pr_{lk} t_{lk}, \tag{3.15}$$

where Pr_{lk} and t_{vk} are the probability and lift travel time of kth scenario respectively.

Obviously, the available lift could be at any tier, but the destination/origin tier cannot be the ground floor because S/R requests on the ground floor need no lift service. Hence,

$$Pr_{lk} = \begin{cases} Pr_s Pr_l Pr_{do} = \frac{\lambda_s}{\lambda_s+\lambda_r} Pr_l Pr_{do} & \text{storage request,} \\ Pr_r Pr_l Pr_{do} = \frac{\lambda_r}{\lambda_s+\lambda_r} Pr_l Pr_{do} & \text{retrieval request,} \end{cases} \tag{3.16}$$

where $Pr_l = 1/T$ is the probability of the available lift position, and Pr_{do} is the probability of the destination/origin position:

$$Pr_{do} = Pr(x = m|m > 1) = \frac{Pr(x = m\&m > 1)}{Pr(m > 1)} = \frac{1}{T-1} / \frac{T-1}{T} = \frac{T}{(T-1)^2}. \tag{3.17}$$

Similar to the vehicle travel time t_{vk}s for different scenarios, the lift travel time t_{lk}s for different scenarios can be shown via Figure 3.6. Here the current position of

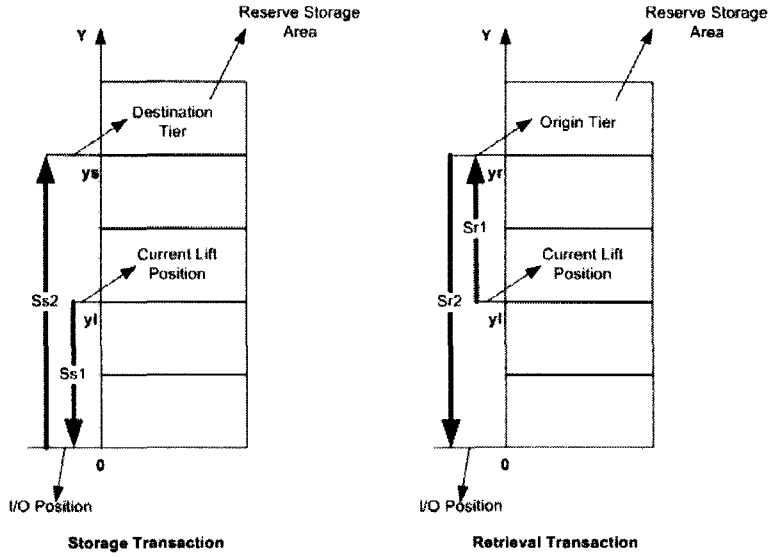

Figure 3.6. Lift travel path for S/R requests

lift is $y_l(m_l)$, the destination tier of a storage request is $y_s(m_s)$, and the origin tier of a retrieval request is $y_r(m_r)$:

$$\begin{aligned} y_l(m_l) &= m_l T_d, m_l = 0, \cdots, T-1, \\ y_s(m_s) &= m_s T_d, m_s = 1, \cdots, T-1, \\ y_r(m_r) &= m_r T_d, m_r = 1, \cdots, T-1, \end{aligned} \tag{3.18}$$

where $T_d = H_p + Z_p$ is the height of each tier.

By considering the acceleration/deceleration of lifts, the travel time t_{ly} between any two tiers s_{ly} is:

$$t_{ly} = \begin{cases} 2\sqrt{\frac{s_{ly}}{a_l}} & s_{ly} < s_{la}, \\ \frac{s_{ly}-s_{la}}{V_l} + 2t_{la} & \text{otherwise}, \end{cases} \tag{3.19}$$

where $s_{la} = t_{al}V_l = 4.5m$, $t_{al} = \frac{V_l}{a_l} = 3s$.

For a storage transaction, the travel path contains two parts, s_{s1} and s_{s2}. s_{s1} is the path for the lift to travel from its current position to the I/O position on the ground floor, and s_{s2} denotes the path for the lift to transfer the pallet from the I/O position to the destination tier.

$$\begin{aligned} s_{s1}(m_l, m_s) &= y_l(m_l) = m_l T_d, \\ s_{s2}(m_l, m_s) &= y_s(m_s) = m_s T_d. \end{aligned} \tag{3.20}$$

The corresponding travel times t_{s1} and t_{s2} can be obtained by equation (3.19) respectively:

$$\begin{aligned} t_{s1}(m_l, m_s) &= \begin{cases} 2\sqrt{\frac{m_l T_d}{a_l}} & m_l T_d < s_{al}, \\ \frac{m_l T_d - s_{al}}{V_l} + 2t_{al} & m_l T_d \geq s_{al}. \end{cases} \\ t_{s2}(m_l, m_s) &= \begin{cases} 2\sqrt{\frac{m_s T_d}{a_l}} & m_s T_d < s_{al}, \\ \frac{m_s T_d - s_{al}}{V_l} + 2t_{al} & m_s T_d \geq s_{al}. \end{cases} \end{aligned} \tag{3.21}$$

Hence, the lift travel time of a storage transaction is: $t_l(S, m_l, h_s) = t_{s1}(m_l, h_s) + t_{s2}(m_l, h_r) + t_2$.

For a retrieval transaction, the travel path also contains two parts: s_{r1} and s_{r2}. s_{r1} is the distance between the lift position and the origin tier of the retrieval request, while s_{r2} denotes the path taken by the lift to transfer the pallet from the origin tier

to the I/O position on the ground floor.

$$\begin{aligned} s_{r1}(m_l, m_r) &= |y_l - y_r| = |m_l - m_r|T_d, \\ s_{r2}(m_l, m_r) &= y_r = m_r T_d. \end{aligned} \tag{3.22}$$

The corresponding travel times t_{r1} and t_{r2} can be obtained by equation (3.19) respectively:

$$\begin{aligned} t_{r1}(m_l, m_r) &= \begin{cases} 2\sqrt{\frac{|m_l - m_r|T_d}{a_l}} & |m_l - m_r|T_d < s_{al}, \\ \frac{|m_l - m_r|T_d - s_{al}}{V_l} + 2t_{al} & |m_l - m_r|T_d \geq s_{al}. \end{cases} \\ t_{r2}(m_l, m_r) &= \begin{cases} 2\sqrt{\frac{m_r T_d}{a_l}} & m_r T_d < s_{al}, \\ \frac{m_r T_d - s_{al}}{V_l} + 2t_{al} & m_r T_d \geq s_{al}. \end{cases} \end{aligned} \tag{3.23}$$

Hence, the lift travel time for a retrieval transaction is: $t_l(R, m_l, m_s) = t_{r1}(m_l, m_r) + t_{r2}(m_l, m_r) + t_2$.

3.4.2. AS/RS parameters

The unit load AS/RS we study here is from TGW-ERMANCO (2007). Details of this warehouse are shown in tables 3.6 and 3.7.

Table 3.6. AS/RS rack system data

Height (H)	$36m$	$12ft$
Level (L)	15	
Aisle (A)	10	
Pallet height (H_p)	$2000mm$	$6.56ft$
Distance between level-pallet (Z_p)	$350mm$	$1.15ft$
Width of a storage position (uL)	$1002mm$	$3.00ft$
Distance between two aisles (D_a)	$4256mm$	$13.96ft$

Table 3.7. AS/RS automated devices data

Crane horizontal speed (V_H)	$4.444m/s$	$13.333ft/s$ $(800ft/min)$
Crane horizontal acceleration/deceleration (a_H)	$4.333m/s^2$	$13ft/s^2$
Crane vertical speed (V_V)	$2.222m/s$	$6.667ft/s$ $(400ft/min)$
Crane horizontal acceleration/deceleration (a_V)	$2.167m/s^2$	$6.5ft/s^2$
Conveyor speed (V_C)	$0.3m/s$	$0.9ft/s$ $(54ft/min)$

The crane travel time within each aisle can be determined by considering three factors: S/R request type (S or R), current crane position ($x_c(i_c)$, $y_c(j_c)$) and destination/origin position ($x_p(i_p)$, $y_p(j_p)$), where i_c and i_p denote the i_cth column and i_pth column, j_c and j_p denote j_cth level and j_pth level. The number of columns C can be determined by storage capacity n:

$$C = \left\lceil \frac{n}{2 \times A \times L} \right\rceil. \tag{3.24}$$

Hence, there are $2 \times L^2 \times C^2$ possible scenarios for an AS/RS, and the effective crane travel time $\mathbb{E}(t_v)$ is

$$\mathbb{E}(t_c) = \sum_{k}^{2 \times L^2 \times C^2} Pr_{ck} t_{ck}, \tag{3.25}$$

where Pr_{ck} and t_{ck} are the probability and crane travel time of kth scenarios respectively.

Similar to equation (3.5), the Pr_{ck} can be obtained as:

$$Pr_{ck} = \begin{cases} Pr_s \frac{1}{L^2} \frac{1}{C^2} = \frac{\lambda_s/A}{\lambda_s/A+\lambda_r/A} \frac{1}{L^2} \frac{1}{C^2} = \frac{\lambda_s}{\lambda_s+\lambda_r} \frac{1}{L^2} \frac{1}{C^2} & \text{storage request,} \\ Pr_r \frac{1}{L^2} \frac{1}{C^2} = \frac{\lambda_r/A}{\lambda_s/A+\lambda_r/A} \frac{1}{L^2} \frac{1}{C^2} = \frac{\lambda_r}{\lambda_s+\lambda_r} \frac{1}{L^2} \frac{1}{C^2} & \text{retrieval request.} \end{cases} \tag{3.26}$$

Figure 3.7 shows the crane travel paths for an S/R transaction within an aisle. Once again, the layout of an aisle has been simplified as a rectangle shape with grids. Each grid is a possible storage position, where the length is uL and the width is T_d.

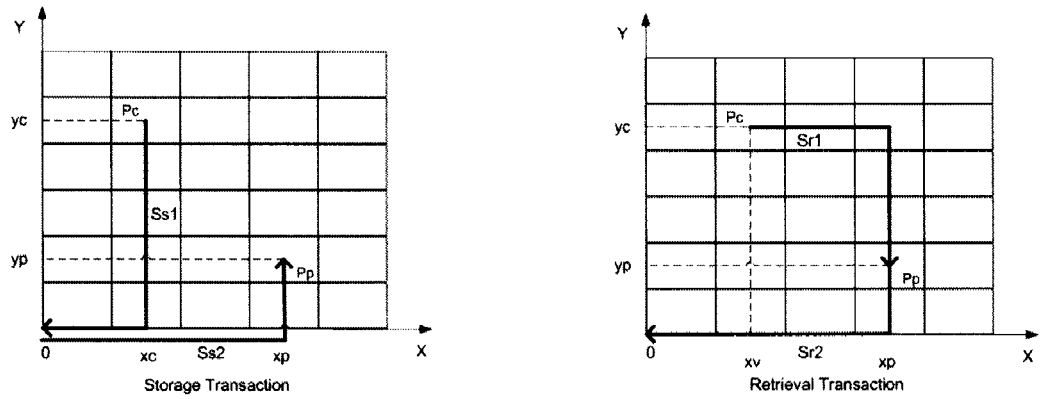

Figure 3.7. Travel path in an aisle for S/R requests

Hence,

$$\begin{aligned} x_c(i_c) &= i_c uL,\ i_c = 0, \cdots, C-1, \\ y_c(j_c) &= j_c T_d,\ j_c = 0, \cdots, L-1, \\ x_p(i_p) &= i_p uL,\ i_p = 0, \cdots, C-1, \\ y_p(j_p) &= j_p T_d,\ j_p = 0, \cdots, L-1. \end{aligned} \tag{3.27}$$

The travel time between any two columns on the x-axis t_x is:

$$t_x = \begin{cases} 2\sqrt{\frac{s_x}{a_H}} & s_x < s_{ax}, \\ \frac{s_x - s_{ax}}{V_H} + 2t_{ax} & \text{otherwise}, \end{cases} \tag{3.28}$$

where $s_{ax} = t_{ax} V_H = 4.56m$, $t_{ax} = \frac{V_H}{a_H} = 1.03s$.

The travel time between any two levels on the y-axis t_y is:

$$t_y = \begin{cases} 2\sqrt{\frac{s_y}{a_V}} & s_y < s_{ay}, \\ \frac{s_y - s_{ay}}{V_V} + 2t_{ay} & \text{otherwise}, \end{cases} \tag{3.29}$$

where $s_{ay} = t_{ay} V_V = 2.28m$, $t_{ay} = \frac{V_V}{a_V} = 1.03s$.

For a storage transaction, the crane travel path has two parts: the path from the current position to the buffer area in front of the conveyor s_{s1}, and the path from the buffer area to the destination position s_{s2}. Here we assume the buffer area of each aisle is at the origin point in figure 3.7.

$$\begin{aligned} s_{s1}(i_c, j_c, i_p, j_p) &= y_c(j_c) + x_c(i_c) = j_c T_d + i_c uL, \\ s_{s2}(i_c, j_c, i_p, j_p) &= x_p(i_p) + y_p(j_p) = i_p uL + j_p T_p. \end{aligned} \tag{3.30}$$

The corresponding travel times t_{s1} and t_{s2} can be obtained from equations (3.28) and (3.29) respectively:

$$t_{s1}(i_c, j_c, i_p, j_p) = \begin{cases} 2\sqrt{\frac{j_c T_d}{a_V}} + 2\sqrt{\frac{i_c uL}{a_H}} & j_c T_d < s_{ay}, i_c uL < s_{ax}, \\ 2\sqrt{\frac{j_c T_d}{a_V}} + \frac{i_c uL - s_{ax}}{V_H} + 2t_{ax} & j_c T_d < s_{ay}, i_c uL \geq s_{ax}, \\ \frac{j_c T_d - s_{ay}}{V_V} + 2t_{ay} + 2\sqrt{\frac{i_c uL}{a_H}} & j_c T_d \geq s_{ay}, i_c uL < s_{ax}, \\ \frac{j_c T_d - s_{ay}}{V_V} + 2t_{ay} + \frac{i_c uL - s_{ax}}{V_H} + 2t_{ax} & j_c T_d \geq s_{ay}, i_c uL \geq s_{ax}, \end{cases}$$

$$t_{s2}(i_c, j_c, i_p, j_p) = \begin{cases} 2\sqrt{\frac{i_p uL}{a_H}} + 2\sqrt{\frac{j_p T_d}{a_V}} & i_p uL < s_{ax}, j_p T_d < s_{ay}, \\ 2\sqrt{\frac{i_p uL}{a_H}} + \frac{j_p T_d - s_{ay}}{V_V} + 2t_{ay} & i_p uL < s_{ax}, j_p T_d \geq s_{ay}, \\ \frac{i_p uL - s_{ax}}{V_H} + 2t_{ax} + 2\sqrt{\frac{j_p T_d}{a_V}} & i_p uL \geq s_{ax}, j_p T_d < s_{ay}, \\ \frac{i_p uL - s_{ax}}{V_H} + 2t_{ax} + \frac{j_p T_d - s_{ay}}{V_V} + 2t_{ay} & i_p uL \geq s_{ax}, j_p T_d \geq s_{ay}. \end{cases} \tag{3.31}$$

The crane travel time for a storage transaction is: $t_c(S, i_c, j_c, i_p, j_p) = t_{s1}(i_c, j_c, i_p, j_p) + t_{s2}(i_c, j_c, i_p, j_p)$.

The crane travel path of a retrieval transaction also includes two parts: the distance between the current position and the origin position of the retrieval request s_{r1}, the distance between the origin position of the retrieval request and the buffer area

s_{r2} (see the right side of Figure 3.7).

$$s_{r1}(i_c, j_c, i_p, j_p) = |x_c(i_c) - x_p(i_p)| + |y_c(j_c) - y_p(j_p)| = |i_c - i_p|uL + |j_c - j_p|T_d,$$

$$s_{r2}(i_c, j_c, i_p, j_p) = x_p(i_p) + y_p(j_p) = i_p uL + j_p T_d. \tag{3.32}$$

The corresponding travel times t_{r1} and t_{r2} can be obtained from equations (3.28) and (3.29) respectively:

$$t_{r1}(i_c, j_c, i_p, j_p) = \begin{cases} 2\sqrt{\frac{|i_c - i_p|uL}{a_H}} + 2\sqrt{\frac{|j_c - j_p|T_d}{a_V}} & |i_c - i_p|uL < s_{ax}, |j_c - j_p|T_d < s_{ay}, \\ 2\sqrt{\frac{|i_c - i_p|uL}{a_H}} + \frac{|j_c - j_p|T_d - s_{ay}}{a_V} + 2t_{ay} & |i_c - i_p|uL < s_{ax}, |j_c - j_p|T_d \geq s_{ay}, \\ \frac{|i_c - i_p|uL - s_{ax}}{V_H} + 2t_{ax} + 2\sqrt{\frac{|j_c - j_p|T_d}{a_V}} & |i_c - i_p|uL \geq s_{ax}, |j_c - j_p|T_d < s_{ay}, \\ \frac{|i_c - i_p|uL - s_{ax}}{V_H} + 2t_{ax} + \frac{|j_c - j_p|T_d - s_{ay}}{a_V} + 2t_{ay} & |i_c - i_p|uL \geq s_{ax}, |j_c - j_p|T_d \geq s_{ay}, \end{cases}$$

$$t_{r2}(i_c, j_c, i_p, j_p) = \begin{cases} 2\sqrt{\frac{i_p uL}{a_H}} + 2\sqrt{\frac{j_p T_d}{a_V}} & i_p uL < s_{ax}, j_p T_d < s_{ay}, \\ 2\sqrt{\frac{i_p uL}{a_H}} + \frac{j_p T_d - s_{ay}}{V_V} + 2t_{ay} & i_p uL < s_{ax}, j_p T_d \geq s_{ay}, \\ \frac{i_p uL - s_{ax}}{V_H} + 2t_{ax} + 2\sqrt{\frac{j_p T_d}{a_V}} & i_p uL \geq s_{ax}, j_p T_d < s_{ay}, \\ \frac{i_p uL - s_{ax}}{V_H} + 2t_{ax} + \frac{j_p T_d - s_{ay}}{V_V} + 2t_{ay} & i_p uL \geq s_{ax}, j_p T_d \geq s_{ay}. \end{cases} \tag{3.33}$$

The crane travel time for a retrieval transaction is: $t_c(R, i_c, j_c, i_p, j_p) = t_{r1}(i_c, j_c, i_p, j_p) + t_{r2}(i_c, j_c, i_p, j_p)$.

Because the conveyor is a continuous server, the effective travel time on a conveyor $\mathbb{E}(t_{cvy})$ is:

$$\mathbb{E}(t_{cvy}) = \sum_{k=1}^{2 \times A} Pr_{cvyk} t_{cvyk}, \tag{3.34}$$

where

$$Pr_{cvyk} = \begin{cases} \frac{\lambda_s}{\lambda_s + \lambda_r} \frac{1}{A} & \text{storage request,} \\ \frac{\lambda_r}{\lambda_s + \lambda_r} \frac{1}{A} & \text{retrieval request,} \end{cases} \tag{3.35}$$

and $t_{cvyk}(a)$ is the travel time between the I/O position and ath aisle:

$$t_{cvyk}(a) = \frac{aD_a}{V_c},\ a = 0, \cdots, A-1. \tag{3.36}$$

3.4.3. Comparison of MPA's performance with simulation

In this set of experiments, we demonstrate the effectiveness of MPA by comparing its estimate of performance measures with those of simulation for several scenarios.

We present two cases of AVS/RS and AS/RS with different throughput requirements. All the simulation results in this thesis are obtained from 100 runs of the simulation model for 1,100 hours with the first 100 hours being the warm-up period. These simulation results are at the 95% confidence level. The performance measures we estimate include the average number of customers L, those in queue in front of lifts and vehicles L_q, and the utilization of each server U.

Table 3.8 shows parameters of the AVS/RS. Table 3.9 shows the comparison between MPA and simulation model for AVS/RS.

Table 3.8. AVS/RS parameters

Aisle	42
Column	36
Tier	7
Vehicle	5 per tier
Lift	7
storage unit	20,000 pallets
t_v	$1.753mins$
t_l	$0.46mins$

Table 3.10 shows parameters of the AS/RS. Table 3.11 shows the comparison between MPA and simulation model for AS/RS.

Our experimental results indicate that the accuracy of MPA is very good when compared to simulation results. Note that the estimates of queue lengths and waiting times provided by MPA are within 2% of those provided by simulation.

Table 3.9. Different throughput requirements (AVS/RS)

High throughput case: $\lambda_s = \lambda_r = 500 pallets/hr$						
	L		L_q		U	
	MPA	**Simu**	**MPA**	**Simu**	**MPA**	**Simu**
Lift	19.094	19.045	12.525	12.476	93.80%	93.80%
Vehicle	7.309	7.352	3.137	3.174	83.40%	83.60%
Normal throughput case: $\lambda_s = \lambda_r = 250 pallets/hr$						
	L		L_q		U	
	MPA	**Simu**	**MPA**	**Simu**	**MPA**	**Simu**
Lift	3.335	3.335	0.051	0.051	46.90%	46.90%
Vehicle	2.136	2.139	0.05	0.05	41.20%	41.80%
Low throughput case: $\lambda_s = \lambda_r = 125 pallets/hr$						
	L		L_q		U	
	MPA	**Simu**	**MPA**	**Simu**	**MPA**	**Simu**
Lift	1.645	1.644	0.001	0.001	23.50%	23.50%
Vehicle	1.046	1.044	0.001	0.001	20.90%	20.90%

Table 3.10. AS/RS parameters

Aisle	10
Level	15
Cranes	10
Storage unit	20,000 pallets
t_{crane}	$0.525 mins$

Table 3.11. Different throughput requirements (AS/RS)

High throughput case: $\lambda_s = \lambda_r = 500 pallets/hr$						
	L		L_q		U	
	MPA	**Simu**	**MPA**	**Simu**	**MPA**	**Simu**
Conveyor	17.726	17.728	0	0		
Crane	8.495	7.076	7.6	6.2	89.50%	87.60%
Normal throughput case: $\lambda_s = \lambda_r = 250 pallets/hr$						
	L		L_q		U	
	MPA	**Simu**	**MPA**	**Simu**	**MPA**	**Simu**
Conveyor	8.874	8.857	0	0		
Crane	0.779	0.778	0.341	0.341	43.80%	43.70%
Low throughput case: $\lambda_s = \lambda_r = 125 pallets/hr$						
	L		L_q		U	
	MPA	**Simu**	**MPA**	**Simu**	**MPA**	**Simu**
Conveyor	4.427	4.428	0	0		
Crane	0.279	0.281	0.061	0.063	21.80%	21.80%

3.4.4. AVS/RS or AS/RS for a given scenario?

In this set of experiments, we use MPA to determine whether to use an AVS/RS or an AS/RS for a given design scenario. The storage capacity of the warehouse is 20,000 pallets and the throughput requirement is: $\lambda_s = \lambda_r = 250\ pallets/hr$.

First, we discuss how to apply AVS/RS to achieve the best result to satisfy the throughput requirement. The AVS/RS configuration considered in table 3.8 can satisfy the storage requirement because

$$A \times 2 \times T \times C = 42 \times 2 \times 7 \times 36 = 21,168 > 20,000,$$

and the reserve storage area parameters can be obtained as:

$$Height = T \times (Z_p + H_p) = 14.5m,$$

$$Width = A \times D_a = 175m,$$

$$Depth = C \times uL = 36m,$$

$$footprint = Width \times Depth = 6,300m^2.$$

According to the result obtained in table 3.9, the utilization of the autonomous devices is quite low. We thus decrease the number of these devices to obtain reasonable levels of utilization as shown in table 3.12.

Table 3.12. AVS/RS resource utilization for alternative configurations

Option		L	L_q	U
V = 5 per tier, L = 7	Lift	3.335	0.051	46.90%
	Vehicle	2.136	0.05	41.20%
V = 3 per tier, L = 7	Lift	3.335	0.051	46.90%
	Vehicle	3.194	1.108	69.50%
V = 5 per tier, L = 4	Lift	6.198	2.914	82.1%
	Vehicle	2.136	0.05	41.20%
V = 3 per tier, L = 4	Lift	6.198	2.914	82.1%
	Vehicle	3.194	1.108	69.50%

If we utilize an AS/RS for this given scenario, based on the experiment illustrated in table 3.10, we see that:

$$A \times 2 \times Level \times C = 10 \times 2 \times 15 \times 67 = 20,100 > 20,000,$$

and the reserve storage area parameters can be obtained as:

$$Height = Level \times (Z_p + H_p) = 35.25m,$$

$$Width = A \times D_a = 42.56m,$$

$$Depth = C \times uL = 67m,$$

$$footprint = Width \times Depth = 2,852m^2.$$

The crane utilization is 43.80% from previous experiments (see table 3.11), which can be improved by adjusting the number of cranes. Because one crane is required for each aisle in the system, changing the number of cranes will change the configuration of entire system. Table 3.13 shows how the warehouse configuration changes when the number of cranes is adjusted.

This set of experiments indicates a potential advantage of the AVS/RS technology in that the reserve storage area and the autonomous devices can be designed separately. If the footprint and the height of a warehouse are fixed, the number of autonomous devices can be adjusted to meet the required throughput with an AVS/RS. Additionally, the number of vehicles and the number of lifts can be set separately as shown in Table 3.12. On the other hand, the AS/RS is a rigid design technology. If the number of cranes is changed, the warehouse needs to be reconfigured extensively.

The throughput requirement of this given scenario is fixed and the AS/RS technology is a better choice. However, the throughput requirement changes frequently in some real applications. For example, the throughput requirement of a food storage

Table 3.13. Performance analysis of alternate AS/RS configurations

Option	Crane = 10	Crane = 8	Crane = 7
A	10	8	7
Level	15	15	15
Column	67	84	96
Height	35.25m	35.25m	35.25m
Width	42.56m	35.25m	29.80m
Depth	67m	84m	96m
footprint	$2,852m^2$	$2,860m^2$	$2,860m^2$
t_{crane}	$0.525mins$	$0.583mins$	$0.625mins$
Conveyor			
L	8.874	6.898	5.908
L_q	0	0	0
Crane			
L	0.779	1.548	2.902
L_q	0.341	0.941	2.159
U	43.80%	60.70%	74.30%

facility changes seasonally. In the summer season, the demand of frozen products, like ice-cream, reaches peak levels. In the winter season, the throughput of frozen products is much lower than the peak. If we use the AS/RS technology to design this warehouse to satisfy the peak level throughput, the utilization of the whole system will be very low during off-peak periods. Some cranes may idle constantly when the throughput level is low.

Obviously, a more flexible warehouse design is to satisfy the required changing throughput levels. The AVS/RS technology is a good choice. According to the previous experiments, we can assign a certain numbers of vehicles and lifts to satisfy the peak level throughput, and remove some autonomous devices to satisfy the lower level throughput. Hence, we can keep the utilization of the whole system at a certain level while the throughput requirement changes.

3.4.5. Zone vs No-zone

Thus far, we have used all the AVS/RS resources to serve the entire warehouse. In the next set of experiments, we divide the warehouse into several zones each with its own set of lifts and vehicles. One set of resources is dedicated to a zone.

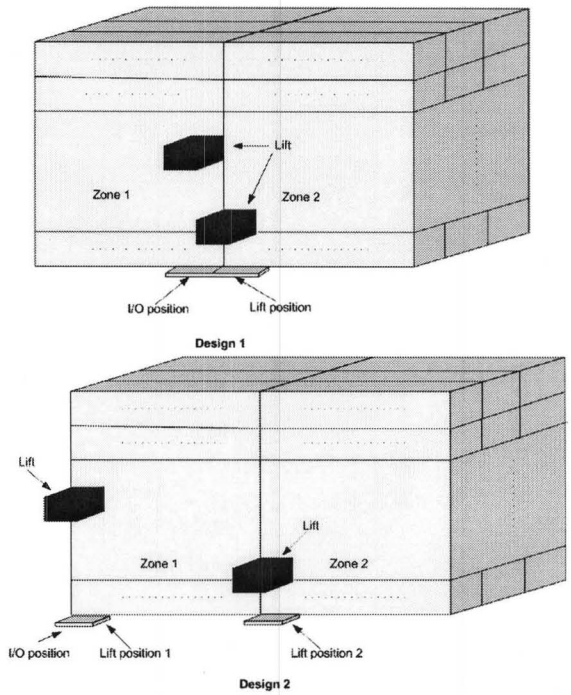

Figure 3.8. Two zone designs

We consider two zone design configurations. In design 1 (see the left side of figure 3.8), the lifts are located near the I/O point as before, and this location is in a corner of the reserve storage area of both zones. In design 2 (see the right side of figure 3.8), the I/O point is at the left corner of the reserve storage area and the lift is at the left corner of each zone. The relevant parameters are shown in table 3.14 and a comparison of the performance of the two zoned designs with that of the unzoned configuration is shown in table 3.15.

Table 3.14. Parameters for the two design configurations in Figure 3.8

Aisle	21
Tier	7
Vehicle	2 per tier
Lift	2
λ_s	$125 pallets/hr$
λ_r	$125 pallets/hr$

Table 3.15. Comparison between the two zone cases and the no-zone case

	L	L_q	U
Zon Design 1			
Lift	8.633	6.851	89.10%
Vehicle	0.818	0.104	35.70%
Zone Design 2			
Lift	8.686	6.904	89.10%
Vehicle	0.818	0.104	35.70%
No-zone			
Lift	6.198	2.916	82.10%
Vehicle	2.299	0.213	52.15%

We see that changing the lift and I/O positions does not have a significant impact on the performance of the alternate design configurations. This suggests that we can consider the lift and I/O position determination separately from the determination of the number of vehicles and lifts. Also, the vehicle utilization in the zoned cases are lower than in the unzoned case. This is due to two reasons. Vehicles travel less in the zoned case and the vehicle queues are separated.

3.5. Conclusions

The AVS/RS with tier-captive vehicles was studied in this chapter. It is compared with the traditional crane-based AS/RS technology, and some advantages of AVS/RS are presented. There are many ways to analyze the AVS/RS and AS/RS. We used the OQN method. MPA, a tool for evaluating OQNs, was chosen to estimate performance measures of AVS/RS and AS/RS. At first, several scenarios were designed to verify the accuracy of MPA by comparing it with a corresponding simulation model. Then,

a series of design questions about AS/RS and AVS/RS were answered. There are more interesting design questions that can also be answered:

- How many input/output (I/O) locations should a high bay area have and what are the optimal locations for a given rack configuration and performance requirement?
- Should the high bay area have allowances for intermediate buffers, and if so, how large should they be and where should they be located? What reductions in cycle times are obtained by introducing such buffers?
- Where should S/R devices idle after processing storage or retrieval transactions? Should they dwell at the point of service completion, in the high-bay area for a storage, and at the I/O points for retrievals?

However, it is not appropriate to use OQN to analyze the system of an AVS/RS with tier-to-tier vehicles. This must be solved by another type of queuing network, SOQN as shown in Chapters 4-6. There is a broad scope for improvement in this area.

CHAPTER 4

PERFORMANCE EVALUATION OF SINGLE-CLASS SOQN

4.1. Introduction

In Chapter 1 we discussed the AVS/RS with tier-to-tier vehicles. In this configuration, a vehicle picks up a load from the staging area near the docks and moves along rails to deposit it in its designated storage position. If the designated position is in another tier, the vehicle interfaces with a lift to reach that tier and again uses rails within that tier to travel in the aisles to reach the designated storage position. Different from AVS/RS with tier-captive vehicles, the vehicle in this configuration travels with the pallet from when it is picked up until it is dropped off, which means it is not suitable to model the problem as an OQN. A better approach is to model the vehicle as an additional resource that must be paired with the pallet and stay with it until the service is completed. Hence, we model the AVS/RS as an SOQN and develop an efficient algorithm to evaluate the performance of the system.

An SOQN represents a queueing network with an additional resource. Initially, all the resources wait in a *resource queue.* An arriving customer is required to be synchronized or paired with a unit of the resource before entering the service network. If there is no resource available, the customer has to wait in an *external queue* until a resource becomes available. Once the customer is synchronized with a resource, the service process begins. When the customer exits the network, the resource associated

with this customer returns to the resource queue and waits for the next arriving customer. A general SOQN is shown in figure 3.1.

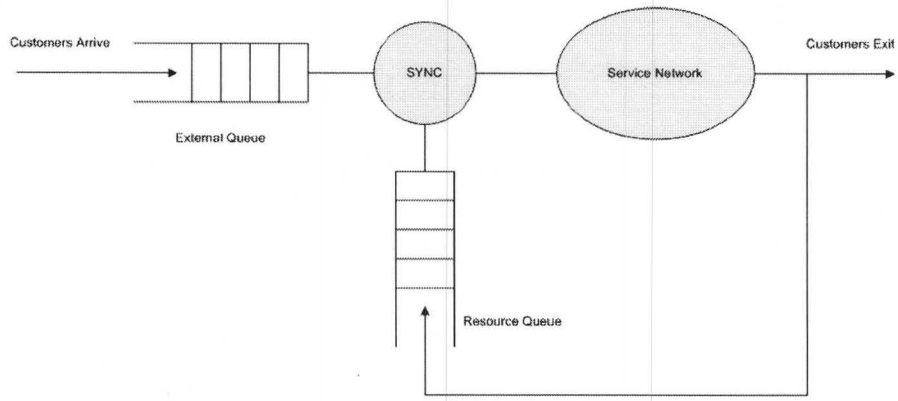

Figure 3.1. SOQN concept

S/R requests on different tiers can be modeled as different classes of customers in the AVS/RS with tier-to-tier vehicles. Additionally, vehicle travel times on different tiers and with the lift are usually generally distributed. As a result, the AVS/RS with tier-to-tier vehicles can be modeled as a multi-class SOQN with generally distributed service times and a general resource. The algorithm to evaluate the performance of multi-class SOQNs is based on the algorithm of single-class SOQNs. Hence, we present two approximate algorithms for single-class SOQN in this chapter.

4.2. SOQN notation

We begin with a two-stage, single-class SOQN that has exponential inter-arrival and service time distributions. We apply two different methods to solve this two-stage, single-class SOQN. The first method involves a solution of state space (Heragu and Srinivasan (2008)) and the second method is based on MGM (Jia and Heragu (2009)). The comparison of these two methods is also given.

After that, we apply the decomposition method and solution of two-stage, single-class SOQN to solve multi-stage, single-class SOQNs.

The main parameters and system performance measures of the AVS/RS (which is modeled as an SOQN) are listed here.

S number of service stages in the network

V number of vehicles in the system

P number of customer classes

λ overall external arrival rate of customers

λ_i external arrival rate of ith class customers, $i = 1, \cdots, P$

μ_j service rate of jth stage, $j = 1, \cdots, S$

L_{eq} average number of customers waiting in the external queue

L_{pq} average number of vehicles in the vehicle pool

L_j average number of customers at jth stage, $j = 1, \cdots, S$

L_n average number of customers in the network

W_s average waiting time per customer in the system

In this thesis, we assume the number of vehicles is known and the route of each class of customers is fixed. The service rate of each server is also assumed to be known and the same for all customers.

4.3. Single-class SOQN with two stages of exponential servers and Poisson arrivals

4.3.1. State space solution

Figure 3.2 shows a two-stage, single-class SOQN. Following Heragu and Srinivasan (2008), the state (i, j) denotes that there are a total of i customers in the external queue and the first server, as well as j customers at the second server. The state space S_s is the infinite set $\{(0,0), (0,1), \cdots, (0,N), (1,0), (1,1), \cdots\}$, and every state

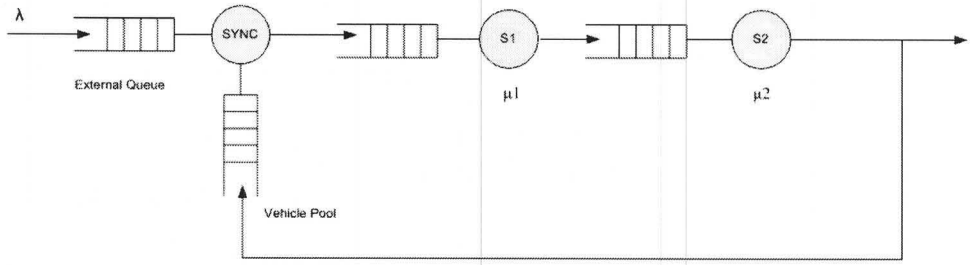

Figure 3.2. Two-stage, single-class SOQN

$s_m \in S_s$ is:

$$s_m = (i, j), \text{ where } i \geq 0, 0 \leq j \leq N \text{ and } m = i(V + 1) + j.$$

Figure 3.3 shows the state space of this SOQN.

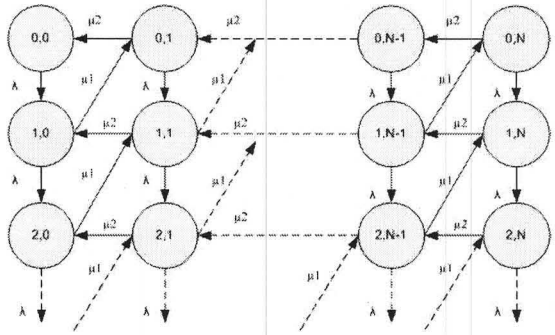

Figure 3.3. The state space of two-stage, single-class SOQN with two variables

This two-stage, single-class SOQN with exponential servers and Poisson arrivals is a CTMC process, which means the conditional *probability mass function* (pmf) of this process satisfies:

$$p_{mn}(t) = P\{X(s + t) = s_m | X(s) = s_n\}, \forall s, t > 0, \text{ and } s_m, s_n \in S_s. \tag{3.1}$$

Here $p_{mn}(t)$ is the transition probability from state s_m to state s_n at time t and $\sum_{s_n \in S_s} p_{mn} = 1, 0 \leq p_{mn} \leq 1$. The p_{mn}s are usually summarized in a nonnegative

transition matrix $\mathbf{P}(t)$:

$$\mathbf{P}(t) = [p_{mn}(t)] = \begin{bmatrix} p_{00}(t) & p_{01}(t) & p_{02}(t) & \cdots \\ p_{10}(t) & p_{11}(t) & p_{12}(t) & \cdots \\ p_{20}(t) & p_{21}(t) & p_{22}(t) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

The unconditional state probability $\pi_n(t)$ can be expressed by $p_{mn}(t)$ and the initial condition $\pi_m(0)$:

$$\pi_n(t) = \sum_{s_m \in S_s} p_{mn}(t)\pi_m(0), \tag{4.2}$$

or

$$\vec{\pi}(t) = \vec{\pi}(0)\mathbf{P}(t), \tag{4.3}$$

where $\vec{\pi}(t) = [\pi_0(t), \pi_1(t), \cdots]$.

The main result of homogeneous CTMCs is the *Kolmogorov's forward differential equation*:

$$p'_{mn}(t) = \sum_{s_k \in S_s} p_{mk}(t)q_{kn}, \tag{4.4}$$

where $q_{mn}(t)$ is the instantaneous transition rate. The definition of q_{mn} is:

$$q_{mn}(t) = \begin{cases} \lim_{\Delta t \to 0} \frac{p_{mn}(t,t+\Delta t)}{\Delta t} & m \neq n, \\ \lim_{\Delta t \to 0} \frac{p_{mm}(t,t+\Delta t)-1}{\Delta t} & \text{otherwise.} \end{cases} \tag{4.5}$$

For example, from state $s_0(0,0)$ to state $s_V(1,0)$, q_{0V} denotes the arrival process of a customer, so $q_{0V} = \lambda$. Since s_0 can only arrive to s_V, the value of q_{00} can be

calculated as:

$$\begin{aligned} q_{00}(t) &= \lim_{\Delta t \to 0} \frac{p_{00}(t+\Delta t) - 1}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{1 - \sum_{s_n \in S_s} p_{0n}(t+\Delta t) - 1}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{-\sum_{s_n \in S_s} p_{0n}(t+\Delta t)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{-p_{0V}(t+\Delta t)}{\Delta t} \\ &= -q_{0V} = -\lambda. \end{aligned}$$

We combine equation (4.2) and (4.4):

$$\dot{\vec{\pi}}(t) = \vec{\pi}(t)\mathbf{Q}, \tag{4.6}$$

where the matrix $\mathbf{Q}$ is:

$$\mathbf{Q} = [q_{mn}], \forall s_m, s_n \in S_s. \tag{4.7}$$

For example, the $\mathbf{Q}$ of the SOQN with two vehicles is:

$$\mathbf{Q} = \begin{bmatrix} -\lambda & 0 & 0 & \lambda & 0 & 0 & 0 & \cdots \\ \mu_2 & -(\mu_2+\lambda) & 0 & 0 & \lambda & 0 & 0 & \cdots \\ 0 & \mu_2 & -(\mu_2+\lambda) & 0 & 0 & \lambda & 0 & \cdots \\ 0 & \mu_1 & 0 & -(\mu_1+\lambda) & 0 & 0 & \lambda & \cdots \\ 0 & 0 & \mu_1 & \mu_2 & -(\mu_1+\mu_2+\lambda) & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \mu_2 & -(\mu_2+\lambda) & 0 & \cdots \\ 0 & 0 & 0 & 0 & \mu_1 & 0 & -(\mu_1+\lambda) & \cdots \\ 0 & 0 & 0 & 0 & 0 & \mu_1 & \mu_2 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

If the unconditional steady state $\vec{\pi}$ of the CTMC exists, it should be independent of time:

$$\lim_{t\to\infty} \dot{\vec{\pi}}(t) = 0.$$

Finally,

$$\vec{\pi}\mathbf{Q} = \mathbf{0}. \tag{4.8}$$

Additionally, the normalization condition holds:

$$\vec{\pi}\mathbf{1} = 1. \tag{4.9}$$

Since the state space of SOQN is infinite, there is no closed form expression for this stochastic process. An alternative method is to truncate the state space at a certain level k to obtain an approximate solution.

Algorithm 4. Algorithm Based on State Space

$\pi_V(0) = 0.5, \vec{\pi}(0) = [0, \cdots, \pi_V(0), \cdots, 0]_{1\times k(V+1)};$

$\vec{\pi}(1) = \vec{\pi(0)}\mathbf{Q}_{k(V+1)\times k(V+1)};$

$n = 0;$

while $|\pi_V(n+1) - \pi_V(n)| \geq \varepsilon$

$n++;$

$\vec{\pi}(n+1) = \vec{\pi}(n)\mathbf{Q}_{k(V+1)\times k(V+1)};$

end

$\vec{\pi} = \vec{\pi}(n+1);$

$\pi_m = \frac{\pi_m}{\sum \pi_m}.$

The performance measures can be obtained directly from these unconditional state probabilities (equations (4.10) - (4.15)).

$$L_{eq} = \sum_{i=0}^{k} \sum_{j=\max(0,V+1-j)}^{V} (i+j-V)\pi_{i(V+1)+j}; \tag{4.10}$$

$$L_1 = \sum_{i=0}^{k}\sum_{j=0}^{V} L_{ij} \text{ where } L_{ij} = \begin{cases} i\pi_{i(V+1)+j} & \text{if } i+j \le V \\ (V-j)\pi_{i(V+1)+j} & \text{otherwise} \end{cases}; \tag{4.11}$$

$$L_2 = \sum_{i=0}^{k}\sum_{j=0}^{V} j\pi_{i(V+1)+j}; \tag{4.12}$$

$$L_n = L_1 + L_2; \tag{4.13}$$

$$L_{pq} = V - L_n; \tag{4.14}$$

$$W_s = \frac{L_n + L_{eq}}{\lambda}. \tag{4.15}$$

4.3.2. Solution based on MGM

As shown above, it is hard to determine the unconditional stationary state probabilities of a Markov process with infinite number of states in a closed form solution. However, if the state space of a Markov process can be expressed by a repetitive structure, the unconditional stationary state probabilities could be obtained exactly. The unconditional stationary state probabilities of this repetitive structure thus has a geometric form. Neuts (1981) developed a body of results of this repetitive structure that is called matrix geometric form. We developed an algorithm based on this MGM to solve the two-stage, single-class SOQN with exponential servers and Poisson arrivals.

First, we construct a state space of this SOQN with three parameters. The first parameter is the number of customers waiting in the external queue i. The second

parameter is the number of customers j at the first server and the last parameter is the number of customers k at the second server.

$$s_m = (i,j,k) \text{ where } i,j,k \geq 0, (j+k) \leq V, \text{ and } m = \begin{cases} \frac{(j+1)j}{2} + k & \text{if } i = 0, \\ i(V+1) + \frac{(N+1)N}{2} + k & \text{otherwise.} \end{cases}$$

The instantaneous transition rates matrix $\mathbf{Q}$ is obtained by equation (4.5). Figure 4.4 shows the state space which can be used to construct the matrix $\mathbf{Q}$.

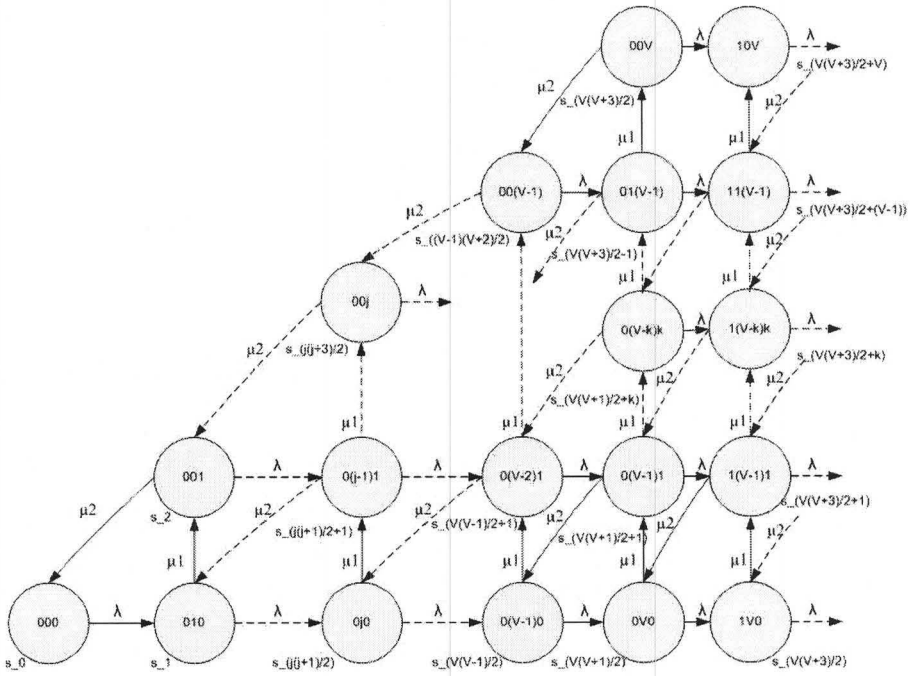

Figure 4.4. The state space of two-stage, single-class SOQN with three variables

Next, we observe the behavior of this Markov process and find the following properties:

1. If $i \geq 1$, $j + k = V$. This property is meaningful because all vehicles are busy if there are customers waiting outside.

2. It is impossible to travel from state (i, j, k) to (i', j, k) when $|i - i'| \geq 2$. Obviously, during an infinitesimal time interval $[t, t + \Delta t]$, only one customer exits or enters the system.

3. In $\mathbf{Q}$, q_{mn}s are independent from i.

Since this Markov process satisfies these properties, it is a continuous time, irreducible, homogeneous *quasi-birth-death* (QBD) process. The original problem now is treated as determining unconditional stationary state probabilities of QBD. In QBD, the number of customers in the external queue i is the ith *level*, and number of customers at each service stage (j, k) is the *phase* (j, k). According to this, we denote $\vec{\pi}_i$ as the vector of unconditional stationary state probabilities of all phases of ith level. This QBD has a repetitive structure of $\mathbf{Q}$ like this:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B_{00}} & \mathbf{B_{01}} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{B_{10}} & \mathbf{A_1} & \mathbf{A_0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{A_2} & \mathbf{A_1} & \mathbf{A_0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{A_2} & \mathbf{A_1} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{4.16}$$

where $\mathbf{B_{00}}$, $\mathbf{B_{01}}$ and $\mathbf{B_{10}}$ are instantaneous transition rate matrixes to determine the initial state of the system.

$\mathbf{B_{00}}$ denotes the transition rates from level 0 to level 0:

$$\begin{bmatrix} -\lambda & \lambda & 0 & & & \\ 0 & -(\mu_1+\lambda) & \mu_1 & \lambda & 0 & \\ \mu_2 & 0 & -(\mu_2+\lambda) & 0 & \lambda & \\ & \ddots & & \ddots & \ddots & \\ & & & 0 \cdots 0 & 0 & -(\mu_1+\lambda) \quad \mu_1 \quad 0 \quad \cdots \quad 0 \\ & & & \mu_2 & & -(\mu_1+\mu_2+\lambda) \quad \mu_1 \\ & & & \vdots & & \vdots \quad \vdots \\ & & & & \mu_2 & -(\mu_1+\mu_2+\lambda) \quad \mu_1 \\ & & & & \mu_2 & -(\mu_2+\lambda) \end{bmatrix}$$

$\mathbf{B_{01}}$ denotes the transition rates from level 0 to level 1:

$$\begin{bmatrix} [\mathbf{0}] \\ \lambda & \\ & \ddots \\ & & \lambda \end{bmatrix}$$

$\mathbf{B_{10}}$ denotes the transition rates from level 1 to level 0:

$$\begin{bmatrix} & 0 & \cdots & 0 & 0 \\ [\mathbf{0}] & \mu_2 & & & \vdots \\ & & \ddots & & \vdots \\ & & & \mu_2 & 0 \end{bmatrix}$$

The repetitive structure includes $\mathbf{A_0}$, $\mathbf{A_1}$ and $\mathbf{A_2}$.

$$\mathbf{A_0} = \begin{bmatrix} \lambda & & & \\ & \lambda & & \\ & & \ddots & \\ & & & \lambda \end{bmatrix}_{(V+1)\times(V+1)},$$

$$\mathbf{A_1} = \begin{bmatrix} -(\mu_1+\lambda) & \mu_1 & & & & \\ & -(\mu_1+\mu_2+\lambda) & \mu_1 & & & \\ & & \ddots & \ddots & & \\ & & & -(\mu_1+\mu_2+\lambda) & \mu_1 & \\ & & & & -(\mu_2+\lambda) & \end{bmatrix}_{(V+1)\times(V+1)},$$

$$\mathbf{A_2} = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \mu_2 & & & \vdots \\ & \ddots & & \vdots \\ & & \mu_2 & 0 \end{bmatrix}_{(V+1)\times(V+1)}.$$

According to equation (4.8), a repeat balance equation holds:

$$\vec{\pi}_{i-1}\mathbf{A_0} + \vec{\pi}_i\mathbf{A_1} + \vec{\pi}_{i+1}\mathbf{A_2} = \mathbf{0},\ i \geq 2. \tag{4.17}$$

The QBD has an important property described in Theorem 7 (Proof can be found in Neuts (1995)).

Theorem 7. *If the QBD is positive recurrent ($\vec{\pi}_i\mathbf{A_0}\vec{e} < \vec{\pi}_i\mathbf{A_2}\vec{e}$), then*

$$\vec{\pi}_{i+1} = \vec{\pi}_i\mathbf{R} \text{ for } i \geq 1, \tag{4.18}$$

or

$$\vec{\pi}_i = \vec{\pi}_1\mathbf{R}^{i-1} \text{ for } i \geq 1, \tag{4.19}$$

where $\mathbf{R}$ *is a rate matrix.*

Substituting equation (4.18) into equation (4.17) and simplifying yields

$$\mathbf{A_0} + \mathbf{RA_1} + \mathbf{R^2A_2} = \mathbf{0}. \tag{4.20}$$

If we can get $\mathbf{R}$ and $\vec{\pi}_1$, we can get all $\vec{\pi}_i$. A simple heuristic procedure is applied to get $\mathbf{R}$. First, the equation (4.20) can be written as

$$\mathbf{R} = -(\mathbf{A_0} + \mathbf{R^2A_2})\mathbf{A_1^{-1}}. \tag{4.21}$$

Then, the procedure to obtain $\mathbf{R}$ is:

$\mathbf{R_0} = \mathbf{0}$

$\mathbf{R_1} = -(\mathbf{A_0} + \mathbf{R_0^2A_2})\mathbf{A_1}$

$k = 0$

while $||\mathbf{R}_{k+1}| - |\mathbf{R}_k|| > \varepsilon$

$k++;$

$\mathbf{R}_{k+1} = -(\mathbf{A_0} + \mathbf{R_k^2 A_2})\mathbf{A_1^{-1}}$;

end

$\mathbf{R} = \mathbf{R_k}$.

$\vec{\pi}_1$ can be obtained from the boundary part of the balance equations (4.8):

$$\begin{cases} \vec{\pi}_1 \mathbf{B_{00}} + \vec{\pi}_1 \mathbf{B_{10}} = \mathbf{0}, \\ \vec{\pi}_0 \mathbf{B_{01}} + \vec{\pi}_1 \mathbf{A_1} + \vec{\pi}_2 \mathbf{A_2} = \mathbf{0}. \end{cases} \tag{4.22}$$

From equation (4.18),

$$\vec{\pi}_2 = \vec{\pi}_1 \mathbf{R}.$$

Substituting this fact into equation (4.22) and simplifying in matrix form, we get:

$$\begin{bmatrix} \vec{\pi}_0 & \vec{\pi}_1 \end{bmatrix} \begin{bmatrix} \mathbf{B_{00}} & \mathbf{B_{01}} \\ \mathbf{B_{10}} & \mathbf{A_1} + \mathbf{R A_2} \end{bmatrix} = \mathbf{0}. \tag{4.23}$$

Since the coefficient matrix is not full rank, equation (4.23) is not sufficient to determine the values of $\vec{\pi}_0$ and $\vec{\pi}_1$. We can use the normalization condition (4.9) to determine these values:

$$1 = \vec{\pi}_0 \vec{e} + \vec{\pi}_1 \sum_{i=1}^{\infty} \mathbf{R}^{i-1} \vec{e} = \vec{\pi}_0 + \vec{\pi}_1 (\mathbf{I} - \mathbf{R})^{-1} \vec{e}. \tag{4.24}$$

Adding equation (4.24) to equation (4.23), we get:

$$\begin{bmatrix} \vec{\pi}_0 & \vec{\pi}_1 \end{bmatrix} \begin{bmatrix} \vec{e} & \mathbf{B_{00}} & \mathbf{B_{01}} \\ (\mathbf{I} - \mathbf{R})^{-1} \vec{e} & \mathbf{B_{10}} & \mathbf{A_1} + \mathbf{R A_2} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0} \end{bmatrix}, \tag{4.25}$$

or

$$\begin{bmatrix} \vec{\pi}_0 & \vec{\pi}_1 \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0} \end{bmatrix} / \begin{bmatrix} \vec{e} & \mathbf{B_{00}} & \mathbf{B_{01}} \\ (\mathbf{I} - \mathbf{R})^{-1} \vec{e} & \mathbf{B_{10}} & \mathbf{A_1} + \mathbf{R A_2} \end{bmatrix}. \tag{4.26}$$

Performance measures can be obtained from these unconditional stationary state probabilities (equation (4.27) - (4.30)):

$$L_{eq} = \sum_{i=1}^{\infty} i\vec{\pi}_i\vec{e} = \vec{\pi}_1(\mathbf{I} - \mathbf{R})^{-2}\vec{e}, \tag{4.27}$$

$$L_n = \vec{n}_0\vec{\pi}_0^T + V\sum_{i=1}^{\infty}\vec{\pi}_i\vec{e} = \vec{n}_0\vec{\pi}_0^T + V\vec{\pi}_1(\mathbf{I} - \mathbf{R})^{-1}\vec{e}, \tag{4.28}$$

$$L_{pq} = V - L_n, \tag{4.29}$$

$$W_s = \frac{L_n + L_{eq}}{\lambda}. \tag{4.30}$$

4.3.3. Numerical experiment

Consider the two-stage, single-class SOQN with two exponential servers. The service rate of first stage μ_1 is 12 and the service rate of the second stage μ_2 is 13. The arrival process is Poisson and the arrival rate λ is 10. We conduct experiments by varying number of vehicles in the system. Results as well as computation times from simulation (**S**), algorithm based on state space (**A1**) and algorithm based on the matrix geometric method (**A2**) are listed in tables 4.1 and 4.2.

Table 4.1. Comparison of **A1** and **S**

	L_{eq}		L_{pq}		L_n		Utilization		W_s		Time	
	A1	**S**	**A1**	**S**	**A1**	**S**	**A1**	**S**	**A1**	**S**	**A1**	**S**
$V = 5$	18.42	19.36	0.38	0.38	4.62	4.62	92.4%	92.4%	146.52	143.78	40.04	27.00
$V = 10$	2.51	2.48	3.40	3.41	6.60	6.59	66.0%	65.9%	54.65	54.48	12.16	27.00
$V = 20$	0.36	0.32	12.01	12.05	7.99	7.95	40.0%	39.8%	50.14	49.63	18.91	27.00
$V = 40$	0.01	0.01	31.61	31.70	8.39	8.33	21.0%	20.8%	40.78	40.18	32.42	27.00

From these results, the algorithm based on state space and the algorithm based on MGM provide estimates of performance measures (e.g., L_{eq}, W_s) that are very close to those of simulation when the utilization of the vehicles is reasonable (say $< 90\%$). When the utilization exceeds 90%, the number of states that must be considered in

Table 4.2. Comparison of **A2** and **S**

	L_{eq}		L_{pq}		L_n		Utilization		W_s		Time	
	A2	**S**	**A2**	**S**	**A2**	**S**	**A2**	**S**	**A2**	**S**	**A2**	**S**
$V = 5$	18.50	19.36	0.38	0.38	4.62	4.62	92.4%	92.4%	138.68	143.78	0.00	27.00
$V = 10$	2.51	2.48	3.40	3.41	6.60	6.59	66.0%	65.9%	54.67	54.48	0.00	27.00
$V = 20$	0.36	0.32	12.01	12.05	7.99	7.95	40.0%	39.8%	50.14	49.63	0.00	27.00
$V = 40$	0.00	0.01	31.82	31.70	8.18	8.33	20.5%	20.8%	49.06	40.18	0.00	27.00

the truncation process increases exponentially. Thus, the algorithm based on state space is not efficient and is either unstable or it takes too long to converge.

4.4. Single-class SOQN with multiple stages of exponential servers and Poisson arrivals

4.4.1. The decomposition-aggregation method

For multiple stages of service, neither the stage-space based method nor a direct application of the MGM are practical. An approximation approach is used to solve this problem. The main idea is to convert the original multi-stage SOQN into an equivalent two-stage SOQN and then apply the algorithms **A1** and **A2** we discussed in Section 4.

First, we combine stages other than the bottleneck stage as a CQN. Then, we apply MVA to solve this CQN to get load-dependent throughput. This CQN can be treated as an equivalent load-dependent server S_e whose service rate is the throughput of this CQN $\mu_e(n)$. Now, the original network can be replaced by a two-stage SOQN where the first stage is the bottleneck stage, and the second stage is a load-dependent server.

This decomposition-aggregation method is based on *Norton's theorem.* Norton's theorem is an important theorem in electrical circuit theory. According to this theorem, the behavior of a subsystem σ between two points is the same when other parts

of this circuit are replaced by a single current source and a parallel internal resistance. The value of the current source equals the current flowing between these two points when the subsystem σ is shorted (Bird (2007)). Chandy et al. (1975) proved that Norton's theorem does hold for queueing networks with local balance. In order to study the behavior of a subsystem σ between two points, other parts can be replaced by a single composite queue. The service rate for this composite queue is equal to the rate at which customers pass between the two points.

Figure 4.5 shows how to apply this method to a multi-stage SOQN. Here we assume the first stage is the bottleneck stage.

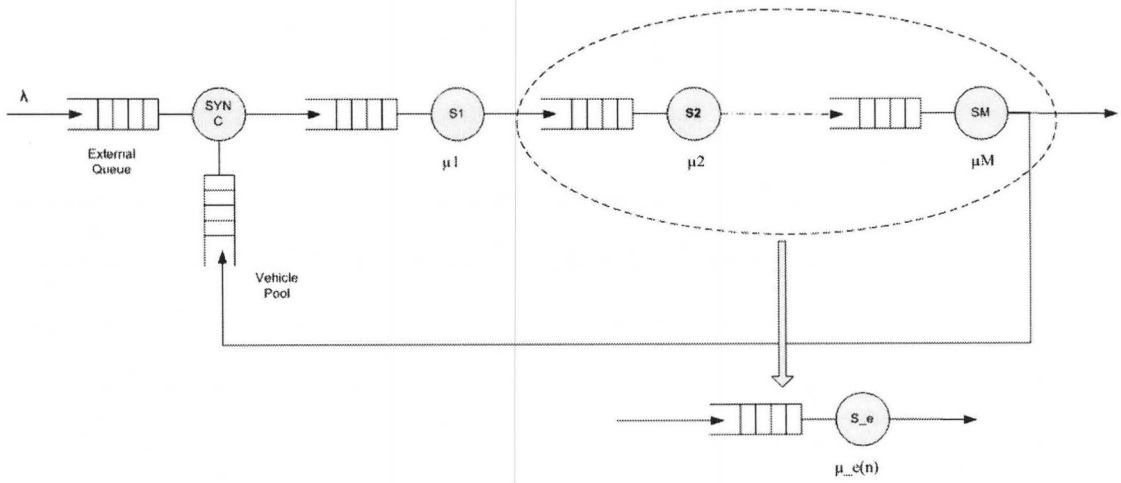

Figure 4.5. Approximation method based on Norton's theorem

4.4.2. Numerical experiment

We conduct a five-stage, single-class SOQN with exponential servers and Poisson arrival. The service rates for these five stages are: $\mu_1 = 12$, $\mu_2 = 13$, $\mu_3 = 15$, $\mu_4 = 14$ and $\mu_5 = 13.5$. The arrival rate λ is 10. As before, we conduct experiments by varying the number of vehicles in the system. Results from simulation (**S**), algorithm

based on state space (**A1**) and algorithm based on matrix geometric method (**A2**) are listed in Tables 4.3 and 4.4.

Table 4.3. Comparison of **A1** and **S**

	L_{eq}		L_{pq}		L_n		Utilization		W_s		Time	
	A1	**S**	**A1**	**S**	**A1**	**S**	**A1**	**S**	**A1**	**S**	**A1**	**S**
$V = 15$	12.02	10.27	1.84	1.83	13.16	13.17	87.7%	87.8%	151.08	140.64	99.34	53.12
$V = 20$	2.80	2.69	5.66	5.62	14.34	14.38	71.7%	71.9%	102.83	102.42	31.37	53.12
$V = 25$	1.02	0.91	10.00	10.02	15.00	14.98	60.0%	59.9%	96.14	95.34	27.68	53.12
$V = 30$	0.41	0.46	14.64	14.56	15.36	15.44	51.2%	51.5%	94.62	95.40	19.62	53.12

Table 4.4. Comparison of **A2** and **S**

	L_{eq}		L_{pq}		L_n		Utilization		W_s		Time	
	A2	**S**	**A2**	**S**	**A2**	**S**	**A2**	**S**	**A2**	**S**	**A2**	**S**
$V = 15$	12.07	10.27	1.83	1.83	13.17	13.17	87.8%	87.8%	151.41	140.64	0.00	53.12
$V = 20$	2.81	2.69	5.66	5.62	14.34	14.38	71.7%	71.9%	102.90	102.42	0.00	53.12
$V = 25$	0.99	0.91	10.02	10.02	14.98	14.98	59.9%	59.9%	95.78	95.34	0.00	53.12
$V = 30$	0.42	0.46	14.45	14.56	15.55	15.44	51.8%	51.5%	86.83	95.40	0.00	53.12

First, the approximation method performs very well, which is indicated by a comparison of the results from the analytical method and those from the simulation model. Second, the algorithm based on state space and the algorithm based on MGM can estimate performance measures of this multi-stage SOQN very well. However, if the utilization of the vehicles is too low (say $< 20\%$), the algorithm based on MGM cannot estimate performance measures accurately. Finally, the algorithm based on state space is not efficient when the utilization of the vehicles is too high (say above 90%).

4.5. Conclusions

In this chapter, we present two approximate algorithms for single-class SOQN with exponentially distributed service times. The first method is the state space

based method. The key point of this method is to truncate the state space of two-stage single-class SOQN at a certain level, then estimate the steady state probabilities. However, if the number of resources is large, this method is time consuming because the state space is too huge to solve. Secondly, the two-stage single-class SOQN is solved by the MGM, which develops a generator matrix with repetitive structures that can be solved exactly via an iterative procedure. From the experimental results, it can be observed that the approximate algorithm based on MGM is faster than the approximate algorithm based on the state space. In the multi-class SOQN with generally distributed service times, we will modify this approximate algorithm based on MGM to estimate performance measures of the network.

CHATPER 5

PERFORMANCE EVALUATION OF GENERALIZED SOQN

5.1. Introduction

In Chapter 3, we discussed how to use MGM to solve single-class SOQNs with exponentially distributed inter-arrival and service times. By using MGM, a set of steady-state results of complicated queueing networks can be obtained via a tractable method and by paying a small price approximation via use of Norton's theorem. This successful application of MGM is due to the unique *lack-of-memory* property of the exponential distribution. However, imposing exponential restrictions on the inter-arrival and service times may not reflect what happens in many real-world situations. On the other hand, analyzing general inter-arrival and service times directly via the simulation method, is highly costly and time-consuming. A compromise is to develop a method that approximates general distributions while using the MGM approach. *Phase-type* distributions is a popular choice for this purpose (Neuts (1995)).

In this chapter, we first introduce the definitions and important properties of phase type distributions. Then, we discuss how to analyze the single class SOQN with generally distributed inter-arrival and service times by using phase type distributions.

5.2. Phase-type distributions

5.2.1. Definition

As stated in Chapter 1, to analyze the property of a random variable S, we usually need first two moments, the mean value of $\mathbb{E}[S]$ and $\mathbb{E}[S^2]$, or the mean $\mathbb{E}[S]$ and the squared coefficient of variation (SCV) of S, $C_X^2(S) = var(S)/(\mathbb{E}[S])^2$. When $C_X^2(S)$ equals 1, S is exponentially distributed. If all random variables of a queueing network model are exponentially distributed, we can analyze it as a Markov process. Otherwise, the queueing network model is a non-Markovian system. Phase-type distributions are useful in approximating a non-Markovian system as a Markovian system. After this approximation process, we can use MGM to analyze this equivalent Markov process.

Erlang (1917) is the earliest paper that introduced the phase concept to approximate general distributions. In this paper, the well-known Erlang-k distribution could be decomposed into k independent and identical exponentially distributions. These k exponential distributions are called k phases of Erlang-k distribution. Figure 4.1 shows a random variable with an Erlang-k distribution.

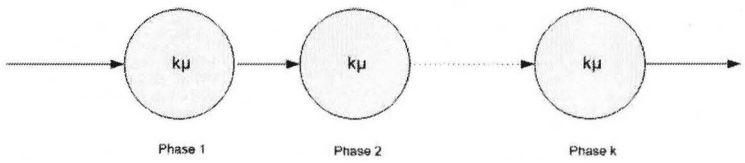

Figure 4.1. A random variable with Erlang-k distribution

Cox (1955) generalized the result of Erlang (1917) and presented the set of phase type distributions (PH-distribution). The definition of PH distributions is given below:

Definition 4. *A probability distribution $F(x)$ is a PH-distribution if and only if the stochastic process of the time until absorption is a finite Markov process Q. The pair $(\alpha, \mathbf{T})$ is a representation of the PH distribution.*

In definition 4, Q is the transition matrix of a finite Markov process with $m+1$ states. States $1 \ldots m$ are transient and absorbed into state $m+1$.

$$Q = \begin{bmatrix} \mathbf{T} & \mathbf{T^0} \\ \mathbf{0} & 0 \end{bmatrix}. \tag{4.1}$$

The distribution $F(x)$ is

$$F(x) = 1 - \alpha \exp(\mathbf{T}x)\mathbf{e}, \ x \geq 0. \tag{4.2}$$

The *Laplace-Stietjes* transform $f(s)$ of $F(x)$ is:

$$f(s) = \mathbb{E}[\exp(-sX)] = \int_{-\infty}^{\infty} e^{-sx} dF(x) = \alpha_{m+1} + \alpha(s\mathbf{I} - \mathbf{T})^{-1}\mathbf{T^0}, \tag{4.3}$$

where the real part of s is bigger than 0.

Additionally, the generator Q^* is $\mathbf{T} + \mathbf{T^0}\mathbf{A^0}$, where $\mathbf{A^0} = (1 - \alpha_{m+1})\mathbf{T^0}\alpha$. Q^* is used to find the stationary probability vector of m states, π:

$$\begin{aligned} \pi\mathbf{Q^*} &= \pi(\mathbf{T} + \mathbf{T^0}\mathbf{A^0}) = \mathbf{0} \\ \pi\mathbf{e} &= 1 \end{aligned} \tag{4.4}$$

The $m \times m$ matrix $\mathbf{T}$ is the transition matrix of m transient states and $\mathbf{T^0}$ is a m transition vector from m transient states to the absorbing state $m+1$. Obviously, T and T^0 satisfy

$$\mathbf{Te} + \mathbf{T^0} = \mathbf{0}, \tag{4.5}$$

where $\mathbf{e}$ is a $m \times 1$ standard unit vector.

The other essential factor to define this Markov process is the initial probability of $m+1$ states, which is given by (α, α_{m+1}). Obviously, α and α_{m+1} should satisfy the following equation:

$$\alpha \mathbf{e} + \alpha_{m+1} = 1. \tag{4.6}$$

From equations (4.5) and (4.6), we can see a pair of $(\alpha, \mathbf{T})$ is enough to represent a PH distribution.

We give two examples to indicate how to define PH distributions. The first example is the classic Erlang-k distribution with parameters $\lambda_1, \ldots, \lambda_k$ and the initial probabilities of the k states are $\alpha = \{1, 0, \ldots, 0\}$. Then, the transition matrix of k states is given by

$$\mathbf{T} = \begin{bmatrix} -\lambda_1 & \lambda_1 & & & & \\ & -\lambda_2 & \lambda_2 & & & \\ & & & \cdots & & \\ & & & & -\lambda_{k-1} & \lambda_{k-1} \\ & & & & & -\lambda_k \end{bmatrix}$$

The transition vector to the absorbed state $k+1$ $\mathbf{T}^0 = -\mathbf{Te}$ is $\{0, \ldots, -\lambda_m\}'$. The initial probability of absorbed state $k+1$ $\alpha_{k+1} = 1 - \alpha\mathbf{e}$ is 0. If $\lambda_1 = \lambda_2 = \ldots = \lambda_k$, C_X^2 of this PH distribution is $1/k$.

The other important PH distribution is the Coxian distribution or Coxian-k distribution. This is also the PH distribution we use in our research. As the name of this distribution indicates, the Coxian-k distribution is represented by a k-phase Markov process. Each phase has an exponentially distributed rate μ_k. After the ith phase, the probability to enter the next phase is a_i, and the probability to be absorbed is b_i. Additionally, $a_i + b_i = 1$. Figure 4.2 shows a variable with Coxian-k distribution.

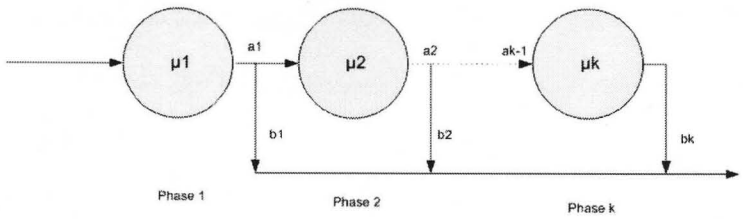

Figure 4.2. Coxian-k distribution

This Coxian-k distribution can be represented by a pair $(\alpha, \mathbf{T})$ where $\alpha = \{1, 0, \ldots, 0\}$ and $\mathbf{T}$ is

$$\begin{bmatrix} -\mu_1 & a_1\mu_1 & & & & \\ & -\mu_2 & a_2\mu_2 & & & \\ & & \ddots & & & \\ & & & -\mu_{k-1} & a_{k-1}\mu_{k-1} & \\ & & & & -\mu_k & \end{bmatrix}.$$

There are two special cases of Coxian-k distribution. The first case is $C_X^2 \leq 1$. In this case, all phases have same service rate μ, and the probability to enter the next phase is 1 except for the first phase. This case is shown in figure 4.3.

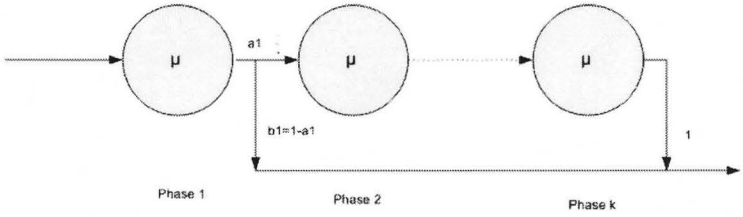

Figure 4.3. Coxian-k distribution with $C_X^2 \leq 1$

The representation of this case is $\alpha = \{1, 0, \ldots, 0\}$ and $\mathbf{T}$ is

$$\begin{bmatrix} -\mu & a_1\mu & & & \\ & -\mu & \mu & & \\ & & \cdots & & \\ & & & -\mu & \mu \\ & & & & -\mu \end{bmatrix}.$$

According to Sauer and Chandy (1981), μ and a_1 can be estimated by equation (4.7),

$$\begin{aligned} \mu &= \frac{k - (1 - a_1)(k-1)}{\bar{X}} \\ a_1 &= 1 - \frac{2kC_X^2 + (k-1) - \sqrt{k^2 + 4 - 4kC_X^2}}{2(C_X^2 + 1)(k-1)}, \end{aligned} \tag{4.7}$$

where $\bar{X}$ is the mean value. The number of phases k can be estimated by equation (4.8),

$$k = \lceil \frac{1}{C_X^2} \rceil. \tag{4.8}$$

The second case is $C_X^2 > 1$. In this case, the number of phase is fixed to 2. Hence, this special Coxian distribution is also called Cox-2 distribution. The service rate of first stage is μ_1 and the service rate of second stage is μ_2. Figure 4.4 shows the Cox-2 distribution.

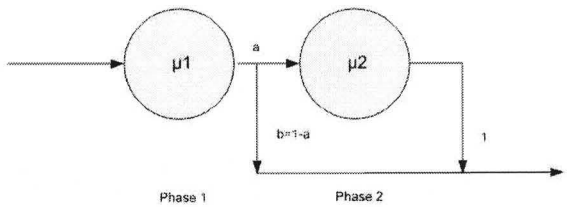

Figure 4.4. Coxian-k distribution with $C_X^2 > 1$ (Cox-2 distribution)

According to Sauer and Chandy (1981), μ_1, μ_2 and a are estimated by equation (5.9),

$$\begin{aligned} \mu_1 &= \frac{2}{\bar{X}} \\ \mu_2 &= \frac{1}{\bar{X}C_X^2} \\ a &= \frac{1}{2C_X^2}. \end{aligned} \tag{5.9}$$

5.2.2. Closure properties and Kronecker product

5.2.2.1. Closure properties. From now on, we can estimate general distributions with different C_X^2s by a PH distribution. It is not that valuable to analyze a single random variable of PH distribution. We start our journey from a single stage queueing model where the inter-arrival and service times are generally distributed. Consider a simple GI/GI/1 queue. Now we can approximate this process as a PH/PH/1 queue, in which the arrival procedure is represented by the pair $(\alpha, \mathbf{T})$ and the service procedure is represented by the pair $(\beta, \mathbf{S})$. How about the behavior of this queue? Do it still hold the PH distribution property? Neuts (1995) proved that the PH distribution property holds even after the mixture.

Theorem 8. *If F is a PH distribution of $m+1$ states with representation $(\alpha, \mathbf{T})$ and G is also a PH distribution of n states with representation $(\beta, \mathbf{S})$, then the convolution $F * G$ is still a PH distribution with representation $(\gamma, \mathbf{L})$, where*

$$\begin{aligned} \gamma &= [\alpha, \alpha_{m+1}\beta] \\ \mathbf{L} &= \begin{bmatrix} \mathbf{T} & \mathbf{T^0B^0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix}. \end{aligned} \tag{5.10}$$

The proof of this theorem is straightforward. The *Laplace-Stietjes* transform of $F * G(x)$ can be expressed as

$$\begin{aligned}
&\gamma_{m+1+n+1} + \gamma(s\mathbf{I} - \mathbf{L})^{-1}\mathbf{L}^{\mathbf{0}} \\
&= \alpha_{m+1}\beta_{n+1} + \beta_{n+1}\alpha(s\mathbf{I} - \mathbf{T})^{-1}\mathbf{T}^{\mathbf{0}} + \alpha_{m+1}\beta(s\mathbf{I} - \mathbf{S})^{-1}\mathbf{S}^{\mathbf{0}} \\
&+ \alpha(s\mathbf{I} - \mathbf{T})^{-1}\mathbf{T}^{\mathbf{0}}\beta(s\mathbf{I} - \mathbf{S})^{-1}\mathbf{S}^{\mathbf{0}} \\
&= [\alpha_{m+1} + \alpha(s\mathbf{I} - \mathbf{T})^{-1}\mathbf{T}^{\mathbf{0}}][\beta_{n+1} + \beta(s\mathbf{I} - \mathbf{S})^{-1}\mathbf{S}^{\mathbf{0}}] \\
&= f(s)g(s).
\end{aligned}$$

Now, we come back to the simple $PH/PH/1$ queue with the arrival procedure $(\alpha, \mathbf{T})$ and the service procedure $(\beta, \mathbf{S})$. According to theorem 8, the distribution of this process is still a PH distribution. Figure 5.5 shows the process of this $PH/PH/1$ queue. Here we assume C_X^2s of both arrival and service procedure are larger than 1.

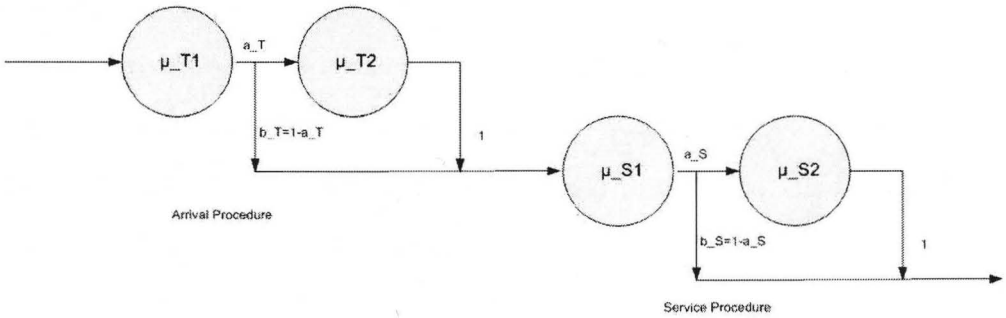

Figure 5.5. A $PH/PH/1$ queue

Next, we analyze states of this $PH/PH/1$ queue. There are 4 stages in this queue. Initially, there is no customer at any stage. Once a customer is generated, there is 1 customer at the first phase of the arrival process. At the next moment, the probability that this customer is transferred to the second phase is a_T, and the probability the customer is absorbed is b_T. Here, the arrival procedure is renewed when the absorption state is reached. At the same time, the customer is transfered

to the first phase of the service process. Hence, we can denote the state of this $PH/PH/1$ queue as (n, a_l, s_l), where n is the number of customers in the service node or the level, a_l is the lth phase of the arrival process and s_l is the lth phase of the service process.

The state space of this $PH/PH/1$ queue is very complicated. However, the Markov process of the $PH/PH/1$ queue can be viewed as an embedded finite Markov process (PH distribution) in a $M/M/1$ queue. The $M/M/1$ queue is an example of the birth-death process. Hence, Neuts (1995) suggested to discuss the $PH/PH/1$ as a direct example of a QBD process. According to the analysis in Section 4, we divide the state space into two parts. The first part is the initial part, or state space between level 0 and level 0, as well as between level 0 and level 1. The second part is the repetitive state space between levels $n-1$ and $n-1$, as well as between levels $n-1$ and n.

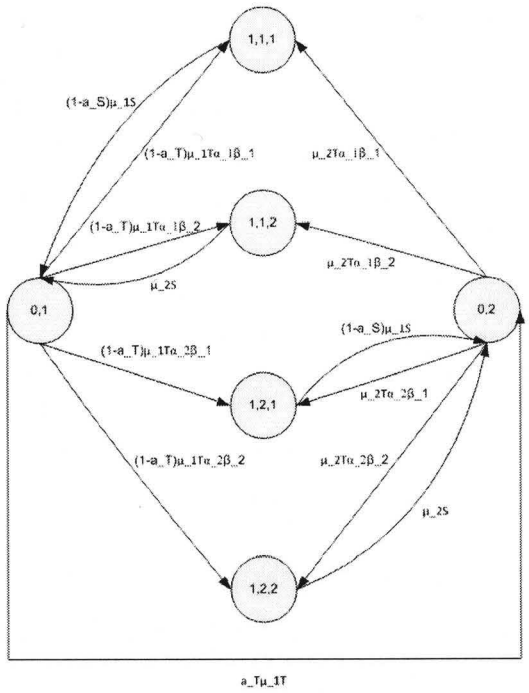

Figure 5.6. State transaction of levels 0 and 1

According to figure 5.6, we can write down the transition matrices of initial part of the state space of this $PH/PH/1$ queue. We use the same notation in Section 4.

$$\mathbf{B_{00}} = \begin{array}{c} \\ (0,1) \\ (0,2) \end{array} \begin{array}{cc} (0,1) & (0,2) \\ -\mu_{1T} & a_T\mu_{1T} \\ 0 & \mu_{2T} \end{array},$$

$$\mathbf{B_{01}} = \begin{array}{c} \\ (0,1) \\ (0,2) \end{array} \begin{array}{cccc} (1,1,1) & (1,1,2) & (1,2,1) & (1,2,2) \\ (1-a_T)\mu_{1T}\alpha_1\beta_1 & (1-a_T)\mu_{1T}\alpha_1\beta_2 & (1-a_T)\mu_{1T}\alpha_2\beta_1 & (1-a_T)\mu_{1T}\alpha_2\beta_2 \\ \mu_{2T}\alpha_1\beta_1 & \mu_{2T}\alpha_1\beta_2 & \mu_{2T}\alpha_2\beta_1 & \mu_{2T}\alpha_2\beta_2 \end{array},$$

$$\mathbf{B_{10}} = \begin{array}{c} \\ (1,1,1) \\ (1,1,2) \\ (1,2,1) \\ (1,2,2) \end{array} \begin{array}{cc} (0,1) & (0,2) \\ (1-a_S)\mu_{1S} & 0 \\ \mu_{2S} & 0 \\ 0 & (1-a_S)\mu_{1S} \\ 0 & \mu_{2S} \end{array}.$$

Similarly, we can get the transition matrices of the repetitive part of the state space of this $PH/PH/1$ queue.

$$\mathbf{A_0} = \begin{array}{c} \\ (n,1,1) \\ (n,1,2) \\ (n,2,1) \\ (n,2,2) \end{array} \begin{array}{cccc} (n-1,1,1) & (n-1,1,2) & (n-1,1,1) & (n-1,1,2) \\ (1-a_T)\mu_{1T}\alpha_1 & 0 & (1-a_T)\mu_{1T}\alpha_2 & 0 \\ 0 & (1-a_T)\mu_{1T}\alpha_1 & 0 & (1-a_T)\mu_{1T}\alpha_2 \\ \mu_{2T}\alpha_1 & 0 & \mu_{2T}\alpha_2 & 0 \\ 0 & \mu_{2T}\alpha_1 & 0 & \mu_{2T}\alpha_2 \end{array},$$

$$\mathbf{A_1} = \begin{array}{c} \\ (n,1,1) \\ (n,1,2) \\ (n,2,1) \\ (n,2,2) \end{array} \begin{array}{cccc} (n,1,1) & (n,1,2) & (n,1,1) & (n,1,2) \\ -\mu_{1T}-\mu_{1S} & a_S\mu_{1S} & a_T\mu_{1T} & 0 \\ 0 & -\mu_{1T}-\mu_{2S} & 0 & a_T\mu_{1T} \\ 0 & 0 & -\mu_{2T}-\mu_{1S} & a_S\mu_{1S} \\ 0 & 0 & 0 & -\mu_{2T}-\mu_{2S} \end{array},$$

$$\mathbf{A_2} = \begin{array}{c} \\ (n-1,1,1) \\ (n-1,1,2) \\ (n-1,2,1) \\ (n-1,2,2) \end{array} \begin{array}{cccc} (n,1,1) & (n,1,2) & (n,1,1) & (n,1,2) \\ (1-a_S)\mu_{1S}\beta_1 & (1-a_S)\mu_{1S}\beta_2 & 0 & 0 \\ \mu_{2S}\beta_1 & \mu_{2S}\beta_2 & 0 & 0 \\ 0 & 0 & (1-a_S)\mu_{1S}\beta_1 & (1-a_S)\mu_{1S}\beta_2 \\ 0 & 0 & \mu_{2S}\beta_1 & \mu_{2S}\beta_2 \end{array}.$$

Now, we can get a similar generator Q as equation (4.16), and apply the MGM to analyze this $PH/PH/1$ queue.

5.2.2.2. Kronecker product. In the previous section, we discussed how to analyze a $GI/GI/1$ queue by applying PH distribution. However, although $PH/PH/1$ is a very simple queue, the generator Q is very complicated. Moreover, theorem 8 can be extended to the convolution of multiple PH distributions. The generator Q of this case will be more complicated and impossible to write down.

Fortunately, an important property of matrices called the *Kronecker product* of matrices can be used to simplify Q. The detail and proof of the *Kronecker product* of matrices can be found in Bellman (1960).

Definition 5. *Let* $\mathbf{A}$ *be an* $m_1 \times n_1$ *matrix and* $\mathbf{B}$ *be an* $m_2 \times n_2$ *matrix. Then the Kronecker product of* $\mathbf{A}$ *and* $\mathbf{B}$, $\mathbf{A} \otimes \mathbf{B}$, *is*

$$\begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n_1}B \\ A_{21}B & A_{22}B & \dots & A_{2n_1}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m_11}B & A_{m_12}B & \dots & A_{m_1n_1}B \end{bmatrix}_{m_1m_2 \times n_1n_2} \tag{5.11}$$

According to Neuts (1995), the generator Q of the $PH/PH/1$ queue can be rewritten as follows:

$$\mathbf{B_{00}} = \mathbf{T}$$

$$\mathbf{B_{01}} = \mathbf{T^0 A^0} \otimes \beta$$

$$\mathbf{B_{10}} = \mathbf{I_T} \otimes \mathbf{S^0}$$

$$\mathbf{A_0} = \mathbf{T^0 A^0} \otimes \mathbf{I_S}$$

$$\mathbf{A_1} = \mathbf{T} \otimes \mathbf{I_S} + \mathbf{I_T} \otimes \mathbf{S}$$

$$\mathbf{A_2} = \mathbf{I_T} \otimes \mathbf{S^0 B^0}.$$

Here $\mathbf{I_T}$ is the diagonal matrix with the same size of $\mathbf{T}$ and $\mathbf{I_S}$ is the diagonal matrix with the same size of $\mathbf{S}$.

5.3. Single-class SOQN with two stages of general servers and general arrivals

5.3.1. State space analysis

In section 4, we applied MGM to analyze a single class SOQN with two stages of exponential servers. Now, we can extand the previous discussion to analyze the single class SOQN with two general service stages. Compared to the $PH/PH/1$ queue we discussed in section 5, the second stage brings new phases. In order to simplify the

notation, we still assume C_X^2 of the service process at the second stage is larger than 1. It is easy to extend to the $C_X^2 \leq 1$ case. Figure 5.7 shows this two-stage SOQN with PH distributions.

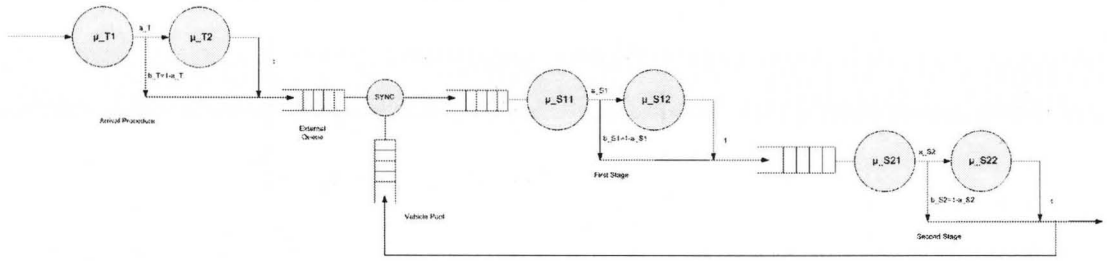

Figure 5.7. A two-stage SOQN with PH distributions

The arrival process is represented by the pair $(\alpha, \mathbf{T})$. The service process at the first stage is represented by the pair $(\beta, \mathbf{S_1})$ and the service process at the second stage is represented by the pair $(\nu, \mathbf{S_2})$. Since we do not want to these processes to begin in the absorption phase, we set $\alpha_3 = \beta_3 = \nu_3 = 0$. Hence,

$$\mathbf{T^0 A_T^0} = \mathbf{T^0}\alpha$$

$$\mathbf{S_1^0 A_{S_1}^0} = \mathbf{S_1^0}\beta$$

$$\mathbf{S_2^0 A_{S_2}^0} = \mathbf{S_2^0}\nu.$$

.

According to the discussion of MGM in section 4, the state of the SOQN with exponentially distributed arrival and service processes is denoted as (i, j), where i is the number of customers at the external queue and the first stage and j is the number of customers at the second stage. We extended this notation and notation of the $PH/PH/1$ queue in section 5 to describe the SOQN with PH distributed arrival and service processes. Each state s_m in the state space $(i, j, a_l, s_{1l}, s_{2l})$ denotes that there are i customers at the external queue and first queue, or level i, there are j customers at the second queue, the current phase of arrival process is a_l and the

current phases of the two service processes are s_{1l} and s_{2l} respectively:

$$s_m = (i, j, a_l, s_{1l}, s_{2l}) \ \textit{where}\ i \geq 0, 0 \leq j \leq N.$$

Similar to the $PH/PH/1$ queue, the Markov process of this SOQN can be viewed as a QBD process with several embedded finite state Markov processes. We can analyze the behavior of this process in the framework of QBD in Section 4. The generator Q of this process are similar to equation (4.16), but much more elaborate. As before, we analyze the initial part and repeated part separately.

$\mathbf{B_{00}}$ is the transition matrix of level 0, where j is changed from 1 to N. This transition matrix can be viewed as a part of the generator of $PH/PH/1$ queue of first N levels. The only difference is that it is impossible to travel from j to $j+1$. This is reasonable because if there is no customer at the external queue and first stage, the number of customers at second stage cannot be increased. This slight difference does not hurt the QBD property. $\mathbf{B_{00}}$ still contains the initial part and repetitive part. In $\mathbf{B_{00}}$, $(0,0)$ denotes two states $(0,0,1,\square,\square)$ and $(0,0,2,\square,\square)$. $(0,j)$ denotes four states $(0,0,1,\square,1)$, $(0,0,1,\square,2)$, $(0,0,2,\square,1)$ and $(0,0,2,\square,2)$. $\square$ means states of this process does not change in this part.

$$\mathbf{B_{00}} = \begin{array}{c|cccccc} & (0,0) & (0,1) & (0,2) & \cdots & & (0,N) \\ (0,0) & \mathbf{T} & & & & & \\ (0,1) & \mathbf{I_T \otimes S_2^0} & \mathbf{T \otimes I_{S_2} + I_T \otimes S_2} & & & & \\ (0,2) & & \mathbf{I_T \otimes S_2^0 \gamma} & \mathbf{T \otimes I_{S_2} + I_T \otimes S_2} & & & \\ \vdots & & & \ddots & \ddots & & \\ (0,N) & & & & & \mathbf{I_T \otimes S_2^0 \gamma} & \mathbf{T \otimes I_{S_2} + I_T \otimes S_2} \end{array}.$$

$\mathbf{B_{01}}$ is the transition matrix from level 0 to level 1, where j is changed from 1 to N. In this part, the situation is more complicated than $\mathbf{B_{00}}$. The initial part is from

$(0,0)$ to $(1,0)$. $(1,0)$ denotes four states $(1,0,1,1,\square)$, $(1,0,1,2,\square)$, $(1,0,2,1,\square)$ and $(1,0,2,2,\square)$.

The transition matrix from $(0,j)$ to $(1,j)$ is different because it involves three PH distributed processes. $(1,j)$ denotes eight states: $(1,j,1,1,1)$, $(1,j,1,1,2)$, $(1,j,1,2,1)$, $(1,j,1,2,2)$, $(1,j,2,1,1)$, $(1,j,2,1,2)$, $(1,j,2,2,1)$ and $(1,j,2,2,2)$. Neuts (1995) proved that theorem 8 can be extended to a more general conclusion: a finite mixture of PH distributions is still a PH distribution. Hence, the transition matrix from $(0,1)$ to $(1,1)$ can be extended from the transition matrix of $PH/PH/1$ queue from level 0 to level 1.

The second difference is the last part of $\mathbf{B_{01}}$ from $(0,N)$ to $(1,N)$. $(1,N)$ denotes four states $(1,N,1,\square,1)$, $(1,N,1,\square,2)$, $(1,N,2,\square,1)$ and $(1,N,2,\square,2)$. Since there are at most N customers at two stages and the number of customer at the second stage is N, the number of customer at the first stage must be 0. Hence, this one customer of $(1,N)$ must be at the external queue waiting for the next available resource. As a result, the states of service process at the first stage do not change.

$$\mathbf{B_{01}} = \begin{array}{c|ccccc} & (1,0) & (1,1) & (1,2) & \cdots & (1,N) \\ (0,0) & \mathbf{T^0}\alpha\otimes\beta & & & & \\ (0,1) & & \mathbf{T^0}\alpha\otimes\beta\otimes\mathbf{I_{S_2}} & & & \\ (0,2) & & & \mathbf{T^0}\alpha\otimes\beta\otimes\mathbf{I_{S_2}} & & \\ \vdots & & & & \ddots & \\ (0,N) & & & & & \mathbf{T^0}\alpha\otimes\mathbf{I_{S_2}} \end{array} .$$

$\mathbf{B_{10}}$ is the transition matrix from level 1 to level 0, where j is changed from 1 to N. The initial part is the transition matrix from $(1,0)$ to $(0,1)$. Similar to $\mathbf{B_{01}}$, the convolution of three PH distributions is still a PH distribution. Hence, the initial part is the mixture of the initial part from level 1 to level 0 at the external queue,

the first stage and the initial part from level 0 to level 1 at the second stage.

$$\mathbf{B_{10}} = \begin{array}{c|cccccc} & (0,0) & (0,1) & (0,2) & \dots & (0,N) \\ (1,0) & & \mathbf{I_T \otimes S_1^0 \otimes \gamma} & & & \\ (1,1) & & & \mathbf{I_T \otimes S_1^0 \otimes I_{S_2}} & & \\ \vdots & & & & \ddots & \\ (1,N-1) & & & & & \mathbf{I_T \otimes S_1^0 \otimes I_{S_2}} \\ (1,N) & & & & & \end{array}.$$

According to Section 4, the repetitive part can also be separated into three parts. $\mathbf{A_0}$ is the transition matrix from level $i-1$ to level i, where j is changed from 1 to N. Obviously, $\mathbf{A_0}$ has a layout similar to $\mathbf{B_{01}}$. The transition matrix of the service process at the second stage is the same in $\mathbf{A_0}$ and $\mathbf{B_{01}}$. Note that although $\mathbf{A_0}$ and $\mathbf{B_{01}}$ look similar, $\mathbf{A_0}$ is the repetitive part and $\mathbf{B_{01}}$ is the boundary part of the generator.

$$\mathbf{A_0} = \begin{array}{c|cccccc} & (i,0) & (i,1) & (i,2) & \dots & (i,N) \\ (i-1,0) & \mathbf{T^0 \alpha \otimes I_{S_1}} & & & & \\ (i-1,1) & & \mathbf{T^0 \alpha \otimes I_{S_1} \otimes I_{S_2}} & & & \\ (i-1,2) & & & \mathbf{T^0 \alpha \otimes \beta \otimes I_{S_2}} & & \\ \vdots & & & & \ddots & \\ (i-1,N) & & & & & \mathbf{T^0 \alpha \otimes I_{S_2}} \end{array}.$$

$\mathbf{A_1}$ is the transition matrix from level i to level i, where j is changed from 1 to N. $\mathbf{A_1}$ should have a layout similar to $\mathbf{B_{00}}$. In $\mathbf{B_{00}}$, the states of the service processes at the first stage do not change. However, $\mathbf{A_1}$ is more complicated because of the

mixture of three PH distributed processes.

$$\mathbf{A_1} = \begin{array}{c|cccccc} & (i,0) & (i,1) & (i,2) & & \cdots \\ (i,0) & \mathbf{T}\otimes\mathbf{I}_{\mathbf{S_1}}+\mathbf{I}_{\mathbf{T}}\otimes\mathbf{S_1} & & & & \\ (i,1) & \mathbf{I}_{\mathbf{T}}\otimes\mathbf{I}_{\mathbf{S_1}}\otimes\mathbf{S_2^0} & (\mathbf{T}\otimes\mathbf{I}_{\mathbf{S_1}}+\mathbf{I}_{\mathbf{T}}\otimes\mathbf{S_1})\otimes\mathbf{I}_{\mathbf{S_2}}+\mathbf{I}_{\mathbf{TS_1}}\otimes\mathbf{S_2} & & & \\ (i,2) & & \mathbf{I}_{\mathbf{T}}\otimes\mathbf{I}_{\mathbf{S_1}}\otimes\mathbf{S_2^0}\gamma & (\mathbf{T}\otimes\mathbf{I}_{\mathbf{S_1}}+\mathbf{I}_{\mathbf{T}}\otimes\mathbf{S_1})\otimes\mathbf{I}_{\mathbf{S_2}}+\mathbf{I}_{\mathbf{TS_1}}\otimes\mathbf{S_2} & & \\ \vdots & & & \ddots & & \ddots \\ (i,N) & & & & & \mathbf{I}_{\mathbf{T}}\otimes\beta\otimes\mathbf{S_2^0}\gamma \\ & (i,N) & & & & \end{array}.$$

$$\begin{array}{c|c} (i,0) & \\ (i,1) & \\ (i,2) & \\ \vdots & \\ (i,N) & \mathbf{T}\otimes\mathbf{I}_{\mathbf{S_2}}+\mathbf{I}_{\mathbf{T}}\otimes\mathbf{S_2} \end{array}$$

$\mathbf{A_2}$ is the transition matrix from level $i+1$ to level i, where j is changed from 1 to N. $\mathbf{A_2}$ has a layout similar to $\mathbf{B_{10}}$. The only difference is that the initial part in $\mathbf{B_{10}}$ should be replaced by the repetitive part in $\mathbf{A_2}$.

$$\mathbf{A_2} = \begin{array}{c|cccccc} & (i,0) & (i,1) & (i,2) & \cdots & (i,N) \\ (i+1,0) & & \mathbf{I}_{\mathbf{T}}\otimes\mathbf{S_1^0}\beta\otimes\gamma & & & \\ (i+1,1) & & & \mathbf{I}_{\mathbf{T}}\otimes\mathbf{S_1^0}\beta\otimes\mathbf{I}_{\mathbf{S_2}} & & \\ \vdots & & & & \ddots & \\ (i+1,N-1) & & & & & \mathbf{I}_{\mathbf{T}}\otimes\mathbf{S_1^0}\otimes\mathbf{I}_{\mathbf{S_2}} \\ (i+1,N) & & & & & \end{array}.$$

From the state space analysis of single-class SOQN with two general stages and arrvial process, we find the generator Q is very complicated. Hence, the state space

solution is not a good choice to solve stationary probabilites. The MGM is used to get stationary probability vectors.

5.3.2. Numerical experiment 1

We conducted two numerical experiments to show the effectiveness of the approximated methods we discussed (**A**). All the results are compared with those from simulation modals (**S**).

The first part is to examine the accuracy of our method for high variance and low variance systems. In Section 5, there are two case of Coxian-k distributions. The first case is a Coxian-k distribution with lower variance and the second case is a Coxian-2 distribution with higher variance. We construct a one stage SOQN or a $PH/PH/1$ queue with population restriction. There are two sets of experiments.

In the first set of experiments, we set the distribution of the inter-arrival time as an exponential distribution with a mean value of 1.5 and the distribution of the service time as an Erlang-2 distribution with a mean value of 1. The exponential distribution is an example of the moderate variance with $C_X^2 = 1$. The Erlang-2 distribution is an example of low variance because C_X^2 of this distribution is 0.5. Similar to the experiments conducted in Section 4, we conduct experiments by varying the number of vehicles V in the system. Table 5.1 shows the number of customers outside L_{eq}, the number of customers at server stage L_{pq} and the utilization of the vehicles.

Table 5.1. Results of Exponential/Erlang-2

	$V = 10$			$V = 5$			$V = 2$		
	L_{eq}	L_{pq}	Utilization	L_{eq}	L_{pq}	Utilization	L_{eq}	L_{pq}	Utilization
A	0.25	3.36	33.6%	0.95	2.66	53.2%	2.12	1.49	74.5%
S	0.19	3.43	34.3%	0.86	2.78	55.6%	1.97	1.64	82.0%
err%	24.0	2.08	1.05	9.47	4.51	5.13	7.08	10.1	29.4

From table 5.1, we can see that our method works well. We can conduct a more radical experiment by assigning higher variant and lower variant distributions to the arrival and service processes respectively. In the second set of experiments, we set the distribution of the inter-arrival time as a Gamma distribution with a mean value of 1.5, C_X^2 1.2 and the distribution of the service time as an Erlang-3 distribution with a mean value of 1, $C_X^2 = 1/3$.

Table 5.2. Results of Gamma/Erlang-3

	$V = 10$			$V = 5$			$V = 2$		
	L_{eq}	L_{pq}	Utilization	L_{eq}	L_{pq}	Utilization	L_{eq}	L_{pq}	Utilization
A	0.009	1.69	16.9%	0.13	1.57	31.4%	0.62	1.08	54.0%
S	0.007	1.93	19.3%	0.11	1.83	36.6%	0.80	1.59	79.5%
err%	22.2	14.2	2.89	15.4	16.6	7.60	12.9	47.2	55.4

As indicated by table 5.2, the accuracy is not as good as the result from table 5.1. One possible reason for the larger error is due to the approximate estimation of the high variance and the low variance distributions.

The second part of our experiment is to examine our approximated method of the two-stage, single-class SOQN with PH distributions. We construct a two-stage SOQN as follows: The inter-arrival time is exponentially distributed with a mean value of 1.5. The distribution of service process at first stage is Erlang-2 distribution with a mean value of 1 and C_X^2 of 0.5. The distribution of the service process at the second stage is a Gamma distribution with a mean value of 1 and C_X^2 of 1.2. We conduct experiments by varying the number of vehicles V in the system.

Table 5.3. Results of two-stage SOQN

	$V = 10$				$V = 5$				$V = 2$			
	L_{eq}	L_1	L_2	Utilization	L_{eq}	L_1	L_2	Utilization	L_{eq}	L_1	L_2	Utilization
A	0.014	0.45	1.54	19.9%	0.18	0.44	1.39	36.5%	1.12	0.38	0.91	64.5%
S	0.012	0.34	1.39	17.3%	0.13	0.33	1.29	32.4%	0.78	0.3	0.88	59.0%
err%	14.3	30.9	9.74	3.24	27.7	25.0	7.2	6.45	30.4	21.1	3.30	15.5

From table 5.3, we can see that the result of the second stage is better than the result of the first stage. It appears that the estimation method of distributions with lower variances needs to be improved. This is one possible direction of future research.

5.3.3. Multiple servers

If there are multiple servers at a service stage, and the service time of each server is exponentially distributed, the service time of the entire stage is no longer exponentially distributed. Neuts (1995) proved that the MGM can give a complete generator of $PH/PH/c$ queue with heterogeneous servers. However, the critical matrix $\mathbf{R}$ used in MGM is hard to compute when the number of parallel servers is large. Hence, in order to get results in reasonable computation time, we make two assumptions on the SOQNs we study. The first assumption is that the servers on the same stage are identical, which means all servers have the same distribution. This assumption allows a major simplification in the description of the state space. This assumption is also reasonable in real application environment. Usually, machines in the same service node execute the same task and should have the same service time distribution. The second assumption is that the number of servers is no larger than 10, which is due to the limitation of the MGM.

The algorithm for the multiple servers situation was first introduced by Mayhugh and McCormick (1968) for the $PH/PH/c$ queue model. Let c_1 and c_2 being the number of parallel servers at the two stages respectively. Each stage s_m now can be described as $(i, j, a_l, s_{1l1}, \ldots, s_{1lc_1}, s_{2l1}, \ldots, s_{2lc_2})$ or $(i, j, a_l, \vec{s_{1l}}, \vec{s_{2l}})$. $\vec{s_{1l}}$ and $\vec{s_{2l}}$ are vectors of current phases of all possible numbers of busy servers at the two stages. It is straightforward to extend the generator of the SOQN of the single-server case to the generator of the SOQN of the multi-server case. We know that if all servers at a stage are busy, the behavior of this stage should be the same as that of the single-server

stage because customers have to wait in the queue in front of that stage. Hence, the only difference is in the initial part when some servers are idle. The framework of the generator Q is rewritten like this:

$$Q = \begin{bmatrix} \mathbf{A_{10}} & \mathbf{A_{00}} & & & & & & & \\ \mathbf{A_{21}} & \mathbf{A_{11}} & \mathbf{A_{01}} & & & & & & \\ & \mathbf{A_{22}} & \mathbf{A_{12}} & \mathbf{A_{02}} & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & \mathbf{A_{2c_1-1}} & \mathbf{A_{1c_1-1}} & \mathbf{A_{0c_1-1}} & & & \\ & & & & \mathbf{A_{2c_1}} & \mathbf{A_{1c_1}} & \mathbf{A_{0c_1}} & & \\ & & & & & \mathbf{A_{2c_1+1}} & \mathbf{A_{1c_1}} & \mathbf{A_{0c_1}} & \\ & & & & & & \ddots & \ddots & \ddots \end{bmatrix}. \tag{5.12}$$

Before we start to analyze this generator, we introduce an additional notation called *Kronecker sum*, which is a simple extension of *Kronecker product* (Definition 5).

Definition 6. *The Kronecker sum of matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *is*

$$\mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes \mathbf{I_B} + \mathbf{I_A} \otimes \mathbf{B}. \tag{5.13}$$

Additionally, The *Kronecker product* and *Kronecker product* of multiple matrices can be expressed as:

$$\begin{aligned} \mathbf{A_0} \otimes \mathbf{A_1} \otimes \ldots \otimes \mathbf{A_N} &= \otimes^N \mathbf{A_n} \\ \mathbf{A_0} \oplus \mathbf{A_1} \oplus \ldots \oplus \mathbf{A_N} &= \oplus^N \mathbf{A_n} \end{aligned} \tag{5.14}$$

In Q, $\mathbf{A_{2t}}$, $\mathbf{A_{1t}}$ and $\mathbf{A_{0t}}$ are extended from $\mathbf{A_2}$, $\mathbf{A_1}$ and $\mathbf{A_0}$ of the single-server case. They indicate the transition behavior when there are t servers busy at the first stage. We choose $\mathbf{A_{1t}}$ to discuss in detail, and give out the result of $\mathbf{A_{0t}}$ and $\mathbf{A_{2t}}$ directly.

Similar to the $\mathbf{A_1}$ of single-server case, $\mathbf{A_{1t}}$ has two parts.

$$\mathbf{A_{1t}} = \begin{array}{c|cccc} & (t,0) & (t,1) & \cdots & (t,V) \\ (t,0) & \mathbf{A_{1t}^{(0,0)}} & & & \\ (t,1) & \mathbf{A_{1t}^{(1,0)}} & \mathbf{A_{1t}^{(1,1)}} & & \\ \vdots & & \ddots & \ddots & \\ (t,V) & & & \mathbf{A_{1t}^{(V,V-1)}} & \mathbf{A_{1t}^{(V,V)}} \end{array}.$$

The first part contains sub-matrices on the diagonal, $\mathbf{A_{1t}^{(v,v)}}$, where v is the number of customers at stage 2. By using definition 6, $\mathbf{A_1^{(v,v)}}$ can be written as $\mathbf{T} \oplus \mathbf{S_1} \oplus \mathbf{S_2}$. Hence,

$$\mathbf{A_{1t}^{(v,v)}} = \mathbf{T} \oplus (\oplus^{\min(t,V-v)}\mathbf{S_1}) \oplus (\oplus^{\min(v,c_2)}\mathbf{S_2}).$$

The second part contains sub-matrices from (t,v) to $(t,v-1)$. In $\mathbf{A_1}$, $\mathbf{A_1^{(1,0)}}$ is $\mathbf{I_T} \otimes \mathbf{I_{S_1}} \otimes \mathbf{S_2^0}$, $\mathbf{A_1^{(v,v-1)}}$ is $\mathbf{I_T} \otimes \mathbf{I_{S_1}} \otimes \mathbf{S_2^0}\gamma$ and $\mathbf{A_1^{(V,V-1)}}$ is $\mathbf{I_T} \otimes \beta \otimes \mathbf{S_2^0}\gamma$. $\mathbf{A_{1t}}$ is more complicated because we must consider different situations of busy servers at second stage.

- $1 \leq v \leq c_2$

In this situation, there is no customer waiting for service at the second stage. Hence, there is no change in the arrival process and the service process at the first stage transactions from (t,v) to $(t,v-1)$.

$$\mathbf{A_{1t}^{(v,v-1)}} = \mathbf{I_T} \otimes (\otimes^t \mathbf{I_{S_1}}) \otimes (\sum_{h=v-1}^{1} \mathbf{S_2^0} \otimes (\otimes^h \mathbf{I_{S_2}}) + \sum_{h=1}^{v-1} (\otimes^h \mathbf{I_{S_2}}) \otimes \mathbf{S_2^0}).$$

- $c_2 \leq v \leq V$ and $t + v \leq V$

In this situation, there are some customers waiting in front of the second stage and no customer is waiting outside. When a customer leaves the system, the first customer in the queue in front of the second stage enters into the stage when the

system transitions from (t, v) to $(t, v-1)$.

$$\mathbf{A_{1t}^{(v,v-1)}} = \mathbf{I_T} \otimes (\otimes^t \mathbf{I_{S_1}}) \otimes (\oplus^{c_2} \mathbf{S_2^0} \gamma).$$

- $c_2 \leq v \leq V$ and $t + v > V$

In this situation, there are customers waiting in front of the second stage and outside. When the system transitions from (t, v) to $(t, v-1)$, a customer leaves the system from the second stage, the first customer in the queue in front of the second stage enters into the stage, and the first customer waiting outside obtains the released resource to start service at the first stage.

$$\mathbf{A_{1t}^{(v,v-1)}} = \mathbf{I_T} \otimes (\otimes^{V-v} \mathbf{I_{S_1}}) \otimes \beta \otimes (\oplus^{c_2} \mathbf{I_{S_2}} \gamma).$$

$\mathbf{A_{0t}}$ is the transition matrix from the current level to the next level, which is similar to $\mathbf{A_0}$ of the single-server case. Hence, $\mathbf{A_{0t}}$ only has sub-matrices on the diagonal.

$$\mathbf{A_{0t}} = \begin{array}{c} \\ (t,0) \\ (t,1) \\ \vdots \\ (t,V) \end{array} \begin{array}{c} \begin{array}{cccc} (t+1,0) & (t+1,1) & \dots & (t+1,V) \end{array} \\ \begin{bmatrix} \mathbf{A_{0t}^{(0,0)}} & & & \\ & \mathbf{A_{0t}^{(1,1)}} & & \\ & & \ddots & \\ & & & \mathbf{A_{0t}^{(V,V)}} \end{bmatrix} \end{array}.$$

- $0 \leq v \leq c_2$

$$\mathbf{A_{0t}^{(v,v)}} = \mathbf{T^0} \alpha \otimes (\otimes^t \mathbf{I_{S_1}}) \otimes \beta \otimes (\otimes^v \mathbf{I_{S_2}}).$$

- $c_2 \leq v \leq V$ and $t + v \leq V$

$$\mathbf{A_{0t}^{(v,v)}} = \mathbf{T^0} \alpha \otimes (\otimes^t \mathbf{I_{S_1}}) \otimes \beta \otimes (\otimes^{c_2} \mathbf{I_{S_2}}).$$

- $c_2 \leq v \leq V$ and $t + v > V$

$$\mathbf{A}_{\mathbf{0t}}^{(\mathbf{v},\mathbf{v})} = \mathbf{T}^{\mathbf{0}}\alpha \otimes (\otimes^{(V-v)}\mathbf{I}_{\mathbf{S_1}}) \otimes (\otimes^{c_2}\mathbf{I}_{\mathbf{S_2}}).$$

$\mathbf{A_{0c_1}}$ is a special case because all servers at the first stage are busy. The next incoming customer has no impact on states of the two stages.

$$\mathbf{A}_{\mathbf{0c_1}}^{(\mathbf{v},\mathbf{v})} = \mathbf{T}^{\mathbf{0}}\alpha \otimes (\otimes^{\min(c_1,V-v)}\mathbf{I}_{\mathbf{S_1}}) \otimes (\otimes^{\min(c_2,v)}\mathbf{I}_{\mathbf{S_2}}).$$

$\mathbf{A_{2t}}$ is the trasition matrix from the current level to the previous level, which is similar to $\mathbf{A_2}$ of the single-server case.

$$\mathbf{A_{2t}} = \begin{array}{c|ccccc} & (t,0) & (t,1) & (t,2) & \dots & (t,V) \\ (t+1,0) & & \mathbf{A}_{\mathbf{2t}}^{(\mathbf{0,1})} & & & \\ (t+1,1) & & & \mathbf{A}_{\mathbf{2t}}^{(\mathbf{1,2})} & & \\ \vdots & & & \ddots & \ddots & \\ (t+1,V-1) & & & & \mathbf{A}_{\mathbf{0t}}^{(\mathbf{V-1,V})} & \\ (t+1,V) & & & & & \end{array}.$$

- $0 \leq v < c_2$

There is no customer waiting outside and in front of the second stage.

$$\mathbf{A}_{\mathbf{2t}}^{(\mathbf{v},\mathbf{v+1})} = \mathbf{I}_{\mathbf{T}} \otimes (\sum_{h=t-1}^{1} \mathbf{S}_{\mathbf{1}}^{\mathbf{0}} \otimes (\otimes^{h}\mathbf{I}_{\mathbf{S_1}}) + \sum_{h=1}^{t-1} (\otimes^{h}\mathbf{I}_{\mathbf{S_1}}) \otimes \mathbf{S}_{\mathbf{1}}^{\mathbf{0}}) \otimes (\otimes^{v}\mathbf{I}_{\mathbf{S_2}}) \otimes \gamma.$$

- $c_2 \leq v \leq V$

In this situation, all servers at the second stage are busy. Customers have to wait in front of the second stage. Hence, the customer who leaves the system will not

change states of the second stage.

$$\mathbf{A}_{\mathbf{2t}}^{(\mathbf{v},\mathbf{v+1})} = \mathbf{I_T} \otimes (\sum_{h=\min(t,V-v)-1}^{1} \mathbf{S_1^0} \otimes (\otimes^h \mathbf{I_{S_1}}) + \sum_{h=1}^{\min(t,V-v)-1} (\otimes^h \mathbf{I_{S_1}}) \otimes \mathbf{S_1^0}) \otimes (\otimes^v \mathbf{I_{S_2}}).$$

$\mathbf{A_{2c_1+1}}$ is special because all servers are busy at the first stage for both the current level and the previous level.

- $0 \leq v < c_2$

In this situation, there is a customer waiting in front of the first stage. When a customer leaves the first stage, the released server begins to serve the waiting customer immediately.

$$\mathbf{A}_{\mathbf{2c_1+1}}^{(\mathbf{v},\mathbf{v+1})} = \mathbf{I_T} \otimes (\oplus^{c_1} \mathbf{S_0}\beta) \otimes (\otimes^v \mathbf{I_{S_2}}) \otimes \gamma.$$

- $c_2 \leq v \leq V - c_1$

In this situation, the first stage is the same as in the previous situation. Customers have to wait in front of the second stage because all the servers at the second stage are busy.

$$\mathbf{A}_{\mathbf{2c_1+1}}^{(\mathbf{v},\mathbf{v+1})} = \mathbf{I_T} \otimes (\oplus^{c_1} \mathbf{S_0}\beta) \otimes (\otimes^{c_2} \mathbf{I_{S_2}}) \otimes \gamma.$$

- $V - c_1 < v \leq V$

All servers at both stages are busy.

$$\mathbf{A}_{\mathbf{2c_1+1}}^{(\mathbf{v},\mathbf{v+1})} = \mathbf{I_T} \otimes (\sum_{h=V}^{1} \mathbf{S_1^0} \otimes (\otimes^h \mathbf{I_{S_1}}) + \sum_{h=1}^{V} (\otimes^h \mathbf{I_{S_1}}) \otimes \mathbf{S_1^0}) \otimes (\otimes^v \mathbf{I_{S_2}}).$$

5.3.4. Numerical experiment 2

We construct a single-class SOQN with two service stages. There are parallel and identical servers in each stage. The distribution of the inter-arrival time is a Gamma

distribution with a mean value of 2 and C_X^2 of 1.2. The first stage has one server, and the distribution of the service time is exponential with a mean value of 1.5. The second stage has 2 parallel servers, and each server has a Erlang-2 distribution for service time with a mean value of 3 and C_X^2 of 0.5.

Similar to the experiments conducted for single-server case, we conduct experiments by varying the number of vehicles V in the system. Table 5.4 shows the number of customers outside L_{eq}, the number of customers at the first stage L_1, the number of customers at the second stage L_2 and the utilization of vehicles.

Table 5.4. Results of two stage SOQN with multiple servers

$V = 10$				
	L_{eq}	L_1	L_2	Utilization
A	0.05	0.91	2.04	29.5%
S	0.07	0.97	1.87	28.3%
err%	40.0	6.59	8.33	4.07
$V = 5$				
A	0.80	0.71	1.85	51.2%
S	0.65	0.67	1.70	47.4%
err%	18.8	5.63	8.33	7.42
$V = 3$				
A	2.58	0.42	1.66	69.2%
S	2.89	0.40	1.45	61.6%
err%	12.0	4.76	12.7	11.0

Results in table 5.4 show that our method is quite accurate for both high variance distributions and low variance distributions.

5.4. Single-class SOQN with multiple stages of general servers and general arrivals

5.4.1. The modified decomposition-aggregation method

In Section 4, we applied a decomposition-aggregation method to reduce a multi-stage SOQN with exponentially distributed arrival and service processes to an equivalent

two-stage SOQN. In this approximated SOQN, one of the two stages is a single load-dependent server, and another one is the bottleneck service stage in the original SOQN. The aggregation method is to construct an equivalent CQN on service stages we choose to aggregate, and use MVA to get the load-dependent throughput.

The same idea is applied here to analyze the multi-stage SOQN with generally distributed arrival and service processes. We can reduce this multi-stage SOQN to an equivalent two-stage SOQN. In this approximated process, the arrival process is PH distributed, one of the two stages is a single-load dependent server with exponentially distributed service process and the remaining one is a multi-server stage with PH distributed service process.

However, the aggregation method needs to be modified. The MVA used to analyze the exponential distribution case cannot be directly applied to the general distribution case. In Chapter 2, we discussed *Marie's Method* (Marie (1980)) to solve non-product-form CQNs. Here, we apply this method to get the load-dependent throughput of the CQN that contains the stages we want to aggregate.

The next modification of the decomposition-aggregation method relates to the representation of the load-dependent exponential distribution as a PH distribution. Then, we can apply the algorithm of the two-stage SOQN with PH distributions to analyze this equivalent two-stage SOQN. One possible solution is to view the exponential distribution as a PH distribution with one transient phase. Let us say $(\alpha, \mathbf{T})$ is the representation of a service stage with a load-dependent exponential distribution. According to equation (5.7), the initial probability of the transient state α is 1, the transition matrix $\mathbf{T}$ is $-\mu(v)$ and the transition matrix of absorbed state $\mathbf{T^0}$ is $\mu(v)$. Here v is the number of customer being served, or the load of this stage.

Figure 5.8 shows the equivalent two-stage SOQN with a PH distributed arrival process, a load-dependent exponentially distributed service stage and a PH distributed service stage.

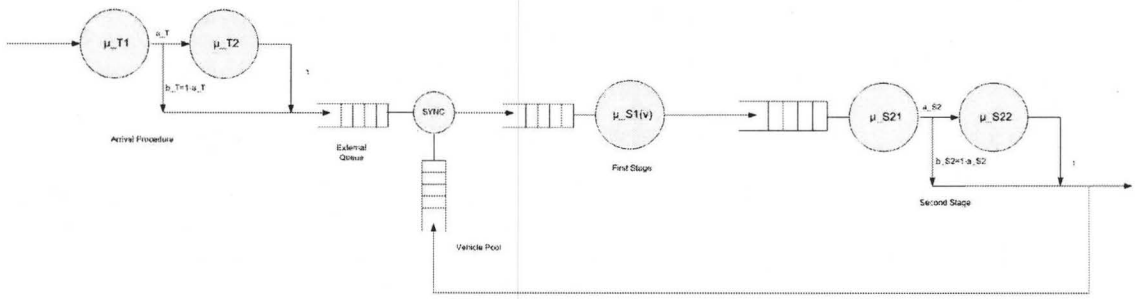

Figure 5.8. The equivalent two-stage SOQN

5.4.2. Numerical experiment 3

The experiment is conducted based on a four-stage single-class SOQN with generally distributed servers and arrival processes. The distribution of the inter-arrival time is a Erlang-2 distribution with a mean value of 1.5. The first stage has a single server with an exponentially distributed service process, where the mean value of service time is 1. The second stage has two identical servers. The distribution of the service time of each server is an Erlang-3 distribution with a mean value of 2. The third stage has three identical servers. The distribution of the service time of each server is a Gamma distribution with a mean value of 3 and C_X^2 of 1.2. The last stage has a single server. The service time has a Gamma distribution with a mean value of 1 and C_X^2 of 2. Table 5.5 shows the configuration of this four-stage SOQN.

The result in table 5.6 shows that our method works well for heavy, normal and lightly loaded network. Again, our method is expected to improve for the low variance distributions. For example, the error is larger when our method is used to estimate the queue length of the second stage with an Erlang-3 distribution.

Table 5.5. Four-stage single-class SOQN

stage i	servers c_i	mean value μ	SCV C_X^2
1	1	1	1
2	2	2	0.33
3	3	3	1.2
4	1	1	2

Table 5.6. Results of four-stage SOQN

$V = 12$					
A	1.02	0.79	0.56	1.20	0.63
S	1.14	0.84	0.50	1.24	0.60
err%	12.1	6.33	10.7	3.33	4.76
$V = 10$					
	L_{eq}	L_1	L_2	L_3	L_4
A	2.31	0.85	0.51	0.92	0.55
S	2.50	0.81	0.45	0.97	0.52
err%	8.23	4.71	11.7	5.43	5.45
$V = 7$					
A	27.3	0.55	0.31	0.43	0.32
S	25.8	0.58	0.28	0.46	0.30
err%	5.50	5.45	9.67	6.98	6.25

5.5. Multi-class SOQN with multiple stages of general servers and general arrivals

There are usually more than one class of customers in real applications of queueing networks. For example, a manufacturing facility needs to process multiple classes of products. Each class of products has its own product routing and process times. Hence, it is valuable to extend the evaluation algorithm of single-class SOQN to multi-class SOQN.

5.5.1. The aggregation method

The algorithm to evaluate multi-class SOQN is inspired by Buitenhek et al. (2000). In this chapter, an aggregation method is presented to evaluate performance measures of a multi-class SOQN. The basic idea is to aggregate multiple classes of customers into

an equivalent class of customers. After this aggregation, we can apply the algorithm of single-class SOQN to get performance measures of this compound class. Finally, we can get performance measures of every single class from the result of single-class SOQN. Results in this chapter show that this aggregation method works very well and the accuracy is good.

The number of customer classes in the multi-class SOQN is denoted by R. The rth class of customer has a generally distributed arrival process with an arrival rate λ_r and the SCV of the inter-arrival time is C_{Xr}^2, where $r = 1, \ldots, R$. Whitt (1983) in his classic paper presented a set of formulae to aggregate multiple arrival processes into a compound arrival process with the arrival rate $\hat{\lambda}$ and the SCV of inter-arrival time $\hat{C}_X^2$ (equation (5.15)).

$$\begin{aligned} \hat{\lambda} &= \sum_{r}^{R} \lambda_r \\ \hat{C}_X^2 &= \sum_{r}^{R} \frac{\lambda_r}{\hat{\lambda}} C_{Xr}^2. \end{aligned} \tag{5.15}$$

We use M to denote the number of service stages. For each class of customers, we assume it has its own and deterministic route, which means a customer cannot change its routing in the network. The server in each stage has different service processes for different classes of customers, the service rate is $c_m \mu_{rm}$ and the SCV of service time is C_{Xrm}^2/c_m, where c_m is the number of parallel servers at the mth stage, $r \in R_m$ and $m = 1, \ldots, M$. Here, R_m is the set of classes of customers who visit the mth stage. Whitt (1983) also presented the set of formulae to aggregate these service processes into one service process for the compound class (equation 5.16).

$$\hat{\mu}_m = \frac{\sum_{r\in R_m} \lambda_r}{\sum_{r\in R_m} \lambda_r / c_m \mu_{rm}}$$
$$\hat{C^2_{Xm}} = \frac{\sum_{r\in R_m} \lambda_r (C^2_{Xrm}/c_m + 1)/(c_m\mu_{rm})^2}{\sum_{r\in R_i} \lambda_r} \hat{\mu}^2_m - 1. \tag{5.16}$$

Each class of customers has its own deterministic path and each station has its own set of classes of customers which visit it. We aggregate these classes into a compound class. However, this compound class is different from the class in the single-class SOQN. In the single-class SOQN, we know the layout of service stages is tandem. In the multi-class SOQN, the compound class may not visit each stage with known probabilities. We use *routing probability* p_{ij} to denote this difference. p_{ij} is the probability that a customer is transferred to the jth stage after service completion at the ith stage. In the single-class SOQN, p_{ij}s are equal to 1 because all stages are in a tandem configuration. In the multi-class SOQN, p_{ij}s may not be 1.

Another important parameter is the *visit ratio* vi_m, which is the mean number of visits of a customer to the mth stage.

$$vi_m = \frac{\hat{\lambda_m}}{\hat{\lambda}}, \tag{5.17}$$

where $\hat{\lambda_m} = \sum_{r\in R_m} \lambda_r$ is the aggregated arrival rate at the mth stage.

The vi_m can also be expressed by routing probabilities,

$$vi_m = \sum_{i=1}^{M} vi_i p_{ji}, \text{ for } i = 1, \ldots, M. \tag{5.18}$$

So far, we have already replaced the original multi-class SOQN with an equivalent single-class SOQN. However, it is still hard to apply the decomposition-aggregation method we used in the single-class SOQN. In the single-class SOQN, we can divide the network into two subnetworks from any node. The average throughput rate of first subnetwork is equal to the arrival rate of the second subnetwork. This fact does

not hold in the multi-class SOQN after aggregation for the reason that each stage has a certain visit ratio and these visit ratios may not be equal to 1. In other words, the throughput rate of the ith node may not be equal to the arrival rate of the jth node. Hence, we cannot divide the network into two parts.

Buitenhek et al. (2000) suggested a simplified decomposition-aggregation method. We simply aggregate all stages and replace it with a load dependent stage. The problem is reduced to solve a simple queue with generally distributed arrival process and a load dependent service stage. We can use the PH distribution we discussed in the previous section to replace the general distribution of the arrival process. Finally, our aim is to solve a $PH/\mu(v)$ queue.

The performance measures of each single class are easy to obtain from the performance measures of the compound class. The external queue length of the rth class of customers is

$$L_{eqr} = L_{eq}\frac{\lambda_r}{\hat{\lambda}}. \tag{5.19}$$

The expected number of customer of rth class at the mth stage L_{mr} could be divided into two parts. The first part is the expected number of customers of rth class in the mth service stage $\rho_{rm} = \frac{\lambda_r}{\mu_{rm}}$. The second part is the expected number of customers of the rth class in front of the mth service stage. It is known that the ratio of the expected number of rth class of customers in the queue should be same as the ratio of the expected number of arrivals of rth class of customers.

$$L_{mr} = \rho_{rm} + (L_m - \sum_{r \in R_m} \rho_{rm})\frac{\lambda_r}{\hat{\lambda}_m}, \text{ for } r \in R_m. \tag{5.20}$$

5.5.2. Numerical experiment 4

We present a series of experiments here to show the accuracy of our method by comparing it with simulation models. We construct an SOQN with six service stages.

There are five different classes of customers in this network. Table 5.7 shows deterministic routes of these five classes.

Table 5.7. Routes of five classes

Class #	Route
1	$S_1 \longrightarrow S_2 \longrightarrow S_5 \longrightarrow S_6$
2	$S_1 \longrightarrow S_4 \longrightarrow S_3 \longrightarrow S_6$
3	$S_2 \longrightarrow S_4 \longrightarrow S_6$
4	$S_1 \longrightarrow S_3 \longrightarrow S_5$
5	$S_2 \longrightarrow S_1 \longrightarrow S_4 \longrightarrow S_3 \longrightarrow S_6 \longrightarrow S_5$

The first set of experiments assumes that arrival processes of five classes are Poisson processes. The arrival rates are 0.6, 0.6, 0.8, 0.8, 1 respectively.

Additionally, the first set of experiments assumes that the six service stages are single server stages. Table 5.8 shows the first two moments of service times of each stage for five classes. The slot with N/A means the particular class does not visit this stage.

Table 5.8. The first two moments of service times

Class #	S_1	S_2	S_3	S_4	S_5	S_6
1	0.2, 2	0.5, 0.8	N/A	N/A	0.4, 1.5	0.25, 0.5
2	0.3, 1.5	N/A	0.3, 1	0.25, 1	N/A	0.25, 0.5
3	N/A	0.3, 1	N/A	0.5, 1	N/A	0.5, 1
4	0.25, 1	N/A	0.4, 1	N/A	0.2, 3	N/A
5	0.15, 0.75	0.26, 1.2	0.2, 1	0.2, 1	0.2, 0.5	0.15, 2

From table 5.8, we can see that there are three kinds of distributions among the service times of six stages: exponential distribution, Erlang-2 distribution and Coxian-2 distribution. These three kinds of distributions represent moderate, low and high variance cases respectively. Similar to the experiments conducted in previous sections, we vary the number of pallets in the network to conduct these experiments from 18 to 25. The expected number of customers at the external queue and each stage for the aggregated class as well as the five classes are shown in tables 5.9 - 5.11.

Table 5.9. Result of 5-class Poisson arrivals 6-stage single server SOQN with 18 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	34.3	1.98	2.55	2.39	3.16	1.77	4.72
A	45.3	2.00	2.50	2.29	2.98	1.77	4.69
err%	24.4	1.00	2.00	4.37	6.04	0.00	0.64
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	5.42	0.39	0.74	N/A	N/A	0.53	0.92
A	7.15	0.39	0.73	N/A	N/A	0.53	0.92
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	5.42	0.45	N/A	0.60	0.75	N/A	0.92
A	7.15	0.45	N/A	0.58	0.71	N/A	0.92
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	7.22	N/A	0.82	N/A	1.20	N/A	1.43
A	9.54	N/A	0.81	N/A	1.14	N/A	1.42
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	7.22	0.55	N/A	0.88	N/A	0.55	N/A
A	9.54	0.56	N/A	0.85	N/A	0.55	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	9.03	0.59	0.99	0.90	1.20	0.69	1.44
A	11.9	0.60	0.97	0.86	1.13	0.69	1.43

Table 5.10. Result of 5-class Poisson arrivals 6-stage single server SOQN with 22 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	8.87	2.04	2.73	2.57	3.24	1.88	5.25
A	9.92	2.06	2.78	2.39	3.16	1.82	5.21
err%	10.6	0.97	1.80	7.53	2.53	3.30	0.77
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.40	0.40	0.78	N/A	N/A	0.56	1.03
A	1.57	0.40	0.79	N/A	N/A	0.55	1.02
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.40	0.46	N/A	0.65	0.77	N/A	1.03
A	1.57	0.46	N/A	0.61	0.75	N/A	1.02
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.87	N/A	0.88	N/A	1.23	N/A	1.57
A	2.09	N/A	0.90	N/A	1.20	N/A	1.56
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.87	0.57	N/A	0.94	N/A	0.57	N/A
A	2.09	0.58	N/A	0.88	N/A	0.57	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.33	0.61	1.06	0.98	1.24	0.73	1.62
A	2.61	0.62	1.09	0.90	1.20	0.71	1.60

Table 5.11. Result of 5-class Poisson arrivals 6-stage single server SOQN with 25 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	4.36	2.09	2.90	2.56	3.25	1.85	5.75
A	4.91	2.09	2.93	2.43	3.24	1.84	5.51
err%	11.2	0.00	1.02	5.35	0.31	0.54	4.00
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.69	0.41	0.83	N/A	N/A	0.55	1.13
A	0.77	0.41	0.83	N/A	N/A	0.55	1.08
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.69	0.47	N/A	0.65	0.78	N/A	1.13
A	0.77	0.47	N/A	0.61	0.77	N/A	1.08
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.92	N/A	0.94	N/A	1.23	N/A	1.70
A	1.03	N/A	0.95	N/A	1.23	N/A	1.64
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.92	0.58	N/A	0.94	N/A	0.58	N/A
A	1.03	0.58	N/A	0.90	N/A	0.57	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.15	0.63	1.14	0.98	1.24	0.72	1.78
A	1.29	0.63	1.15	0.92	1.24	0.72	1.70

Results in these tables show that our approximation method works well when compared to the simulation models. Relative errors of expected number of customers in front of the six stages are very small. Although the relative error of expected number of customers outside is larger in the heavy load case, our method works well for moderate and light load cases.

In the second set of experiments, we examine the accuracy of our method for general arrival processes. We keep the arrival rates of five classes the same, but change some of them to generally distributed processes. Now, the distribution of the arrival processes of class 1 and class 2 are Coxian-2 distributions with $C_X^2 = 2$. The distribution of arrival processes of classes 3 and 4 are still exponential distributions. The distribution of arrival process of class 5 is Erlang-2 distribution. We conduct this set of experiments by changing the number of vehicles in the network from 20 to 25.

Tables 5.12 - 5.14 show the expected number of customers at the external queue and each stage for the aggregated class and the five classes.

Table 5.12. Result of 5-Class general arrivals 6-stage single server SOQN with 20 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	16.8	2.03	3.66	2.33	3.23	1.83	4.93
A	20.7	2.04	3.66	2.35	3.08	1.80	4.98
err%	18.8	0.49	0.00	0.85	4.87	1.67	1.00
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.65	0.40	1.02	N/A	N/A	0.55	0.97
A	3.27	0.40	1.02	N/A	N/A	0.54	0.98
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.65	0.46	N/A	0.59	0.77	N/A	0.97
A	3.27	0.46	N/A	0.59	0.73	N/A	0.98
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	3.54	N/A	1.19	N/A	1.23	N/A	1.49
A	4.36	N/A	1.19	N/A	1.18	N/A	1.50
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	3.54	0.57	N/A	0.86	N/A	0.57	N/A
A	4.36	0.57	N/A	0.87	N/A	0.56	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	4.42	0.61	1.45	0.88	1.23	0.71	1.51
A	5.45	0.61	1.45	0.89	1.17	0.70	1.53

Similar to the first set of experiments, these tables show that our method works very well for estimating expected number of customers in the network. If the load of this SOQN is moderate, the accuracy of our method is also good for estimating the expected number of customers outside.

From the previous two sets of experiments we notice that the queue lengths of stages 2, 4 and 6 are long, which remind us to add additional servers in these stages. Hence, we conduct the last set of experiments by adding some parallel servers. We set the numbers of servers at stage 2 and 4 as 2, and the number of servers at stage 6 as 3. Similarly, we conduct this set of experiments by changing the number of vehicles in the network from 7 to 10. The arrival processes of five classes are still generally distributed and have the same parameters as in the second set of experiments. Tables

Table 5.13. Result of 5-class general arrivals 6-stage single server SOQN with 22 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	10.7	2.07	4.00	2.40	3.18	1.79	5.14
A	11.2	2.07	3.80	2.39	3.17	1.83	5.23
err%	4.30	0.00	5.26	0.42	0.32	2.19	1.72
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.69	0.40	1.10	N/A	N/A	0.54	1.01
A	1.77	0.40	1.05	N/A	N/A	0.55	1.03
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.69	0.46	N/A	0.61	0.76	N/A	1.01
A	1.77	0.46	N/A	0.60	0.76	N/A	1.03
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.26	N/A	1.31	N/A	1.21	N/A	1.54
A	2.36	N/A	1.24	N/A	1.21	N/A	1.57
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.26	0.58	N/A	0.89	N/A	0.56	N/A
A	2.36	0.58	N/A	0.88	N/A	0.57	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	2.82	0.62	1.59	0.91	1.21	0.70	1.58
A	2.95	0.62	1.51	0.90	1.21	0.71	1.61

Table 5.14. Result of 5-class general arrivals 6-stage single server SOQN with 25 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	5.51	2.10	3.98	2.68	3.28	1.81	5.72
A	5.68	2.10	3.96	2.44	3.26	1.85	5.55
err%	3.00	0.00	0.51	9.84	0.61	2.16	3.06
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.87	0.41	1.10	N/A	N/A	0.54	1.12
A	0.90	0.41	1.09	N/A	N/A	0.55	1.09
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.87	0.47	N/A	0.68	0.78	N/A	1.12
A	0.90	0.47	N/A	0.62	0.78	N/A	1.09
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.16	N/A	1.30	N/A	1.24	N/A	1.70
A	1.20	N/A	1.29	N/A	1.24	N/A	1.65
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.16	0.59	N/A	0.98	N/A	0.56	N/A
A	1.20	0.59	N/A	0.90	N/A	0.58	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.45	0.63	1.59	1.03	1.25	0.70	1.77
A	1.49	0.63	1.58	0.93	1.25	0.72	1.72

5.15 - 5.17 show the expected number of customers at the external queue and each stage for the aggregated class and the five classes.

Table 5.15. Result of 5-class general arrivals 6-stage multiple server SOQN with 7 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	36.1	1.66	0.66	1.89	0.62	1.44	0.44
A	30.4	1.64	0.65	1.85	0.61	1.47	0.41
err%	18.8	1.22	1.54	2.16	1.64	2.04	7.32
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	5.70	0.32	0.27	N/A	N/A	0.45	0.07
A	4.80	0.32	0.26	N/A	N/A	0.46	0.06
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	5.70	0.38	N/A	0.48	0.12	N/A	0.07
A	4.80	0.38	N/A	0.47	0.12	N/A	0.06
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	7.60	N/A	0.19	N/A	0.36	N/A	0.29
A	6.40	N/A	0.19	N/A	0.35	N/A	0.28
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	7.60	0.47	N/A	0.72	N/A	0.44	N/A
A	6.40	0.46	N/A	0.70	N/A	0.45	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	8.50	0.49	0.20	0.70	0.15	0.55	0.01
A	8.00	0.48	0.20	0.68	0.14	0.56	0.00

Similar to the first two sets of experiments, these tables show that our method still works well for estimating the expected number of customers in the network. However, the accuracy for estimating the expected number of customers in the external queueing is not very good. There are two sources of error. The first source is from the aggregation process of multiple classes of customers. The second source of error is from the aggregation process of parallel servers.

5.6. Conclusions

In this chapter, we first introduced the concept of PH distribution and some important properties of this distribution. The reason to introduce this distribution is that it can be used to approximate general distribution by the given first two moments

Table 5.16. Result of 5-class general arrivals 6-stage multiple server SOQN with 8 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	7.43	1.76	0.69	1.95	0.64	1.53	0.43
A	9.71	1.73	0.66	1.97	0.62	1.54	0.41
err%	23.5	1.73	4.55	1.02	3.23	0.65	4.88
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.17	0.34	0.27	N/A	N/A	0.47	0.07
A	1.53	0.34	0.27	N/A	N/A	0.48	0.06
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.17	0.40	N/A	0.49	0.12	N/A	0.07
A	1.53	0.40	N/A	0.50	0.12	N/A	0.06
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.56	N/A	0.20	N/A	0.36	N/A	0.29
A	2.04	N/A	0.19	N/A	0.36	N/A	0.28
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.56	0.50	N/A	0.74	N/A	0.47	N/A
A	2.04	0.49	N/A	0.74	N/A	0.47	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	1.96	0.52	0.21	0.72	0.15	0.59	0.01
A	2.56	0.51	0.20	0.73	0.15	0.59	0.00

Table 5.17. Result of 5-class general arrivals 6-stage multiple server SOQN with 10 pallets

Aggregated	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	3.20	1.83	0.69	2.13	0.63	1.72	0.44
A	3.18	1.87	0.67	2.17	0.63	1.65	0.41
err%	0.63	2.14	2.98	1.84	0.00	4.24	7.32
Class 1	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.51	0.36	0.27	N/A	N/A	0.52	0.07
A	0.50	0.36	0.27	N/A	N/A	0.50	0.06
Class 2	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.51	0.42	N/A	0.54	0.12	N/A	0.07
A	0.50	0.42	N/A	0.55	0.12	N/A	0.06
Class 3	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.67	N/A	0.20	N/A	0.36	N/A	0.29
A	0.67	N/A	0.20	N/A	0.36	N/A	0.28
Class 4	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.67	0.51	N/A	0.80	N/A	0.53	N/A
A	0.67	0.53	N/A	0.81	N/A	0.51	N/A
Class 5	L_{eq}	L_1	L_2	L_3	L_4	L_5	L_6
S	0.84	0.54	0.21	0.80	0.15	0.67	0.01
A	0.84	0.56	0.21	0.81	0.15	0.64	0.00

of the distribution. The special structure of the generator matrix of PH distribution makes it a good application of MGM.

Similar to Chapter 4, we started to solve the two-stage SOQN with generally distributed service processes and arrival process. The structure of the generator matrix is discussed in detail. Different from exponentially distributed SOQN, the behavior of generally distributed SOQN with parallel servers is more complicated. We also discussed the structure of the generator matrix in detail, especially how to extend the single-server case to the multi-server case.

Then, we extended the two-stage SOQN to multi-stage SOQN by applying the decomposition-aggregation method we used in the previous chapter. We modified the approximation algorithm by using Marie's method for general distributions to replace the MVA method for the exponential distributions.

Finally, we discussed the approximation algorithm for multiple classes SOQN. The basic idea is to aggregate multiple classes into an aggregated class and then aggregate the network into a single load-dependent stage by using Marie's method. Experiments we conducted show that our method works well for moderately loaded SOQN with low, moderate and high variance distributions.

CHATPER 6

MODEL THE AVS/RS WITH TIER-TO-TIER VEHICLES AS AN SOQN

6.1. Introduction

In Chapter 3, we discussed how to analyze the single-class SOQNs and presented two algorithms to estimate the performance measures for this type of SOQNs. However, single-class SOQNs cannot be applied to model AVS/RSs with tier-to-tier vehicles directly because there are more than one class of customers in AVS/RSs. Thus, AVS/RSs with tier-to-tier vehicles must be modeled as multi-class SOQNs. As discussed in chapter 1, multi-class SOQNs can be aggregated as single-class SOQNs, provided the additional resources to be paired with any class of customers are identical. Hence, solutions of single-class SOQNs serve as the basis for solutions of multi-class SOQNs. In this chapter, we first present how to model the AVS/RS with tier-to-tier vehicles as a multi-class SOQN. Then, we discuss the difference between two synchronization process policies. After that, we conduct a case study of an AVS/RS to demonstrate our method.

6.2. Modeling the AVS/RS as an SOQN

In this section, we discuss how to model the AVS/RS as an SOQN that contains one or more classes of customers, a service network with multiple service stages and an additional set of resources. Similar to Heragu and Srinivasan (2008), we denote the service network as the *network* and the entire system which includes the network,

external queue and resource queue as the *system*. Obviously, the autonomous vehicles are the additional resource. The following subsections describe how to define customers and network of the SOQN in the corresponding AVS/RS.

6.2.1. Customers

First of all, we treat different S/R requests as different classes of customers. The number of S/R request types depends on three criteria. The first criterion is that an S/R request is either a storage or a retrieval request. The second criterion is the request position. This request can occur randomly from anywhere in the system. The travel time from one point to another point can be described as a random variable with a certain distribution. If the request is on the ground floor and the vehicle is also on the ground floor, there is no need for lift service. Therefore, we divide S/R requests based on whether or not it is to/from the ground floor. The last criterion is the available vehicle position. There are four different types of vehicle positions: the input point (**I**), the output point (**O**), the ground floor other than I/O points (**G**) and other tiers (**Ot**).

According to these criteria, the different classes of customers can be expressed by three parameters: **S** or **R** indicating whether the storage or retrieval request; **G** or **Ot** indicating whether the request occurs on the ground floor or on other tiers; the last parameter determines the available vehicle's position. The input point is **I**, while the output point is **O**. The ground floor is **G**, while other tiers are **Ot**s. For example, **(S, Ot, G)** means the storage request is on other tiers and the available vehicle is on the ground floor. Now, there are 16 different types of S/R requests in the AVS/RS, which means there are 16 different classes of customers in the corresponding SOQN.

Here we need to explain the concept of the *synchronization process*. In an SOQN, the merging of a customer and an available resource is called synchronization, which

occurs instantaneously. Accordingly, in AVS/RS, the synchronization process means an available vehicle is assigned to an S/R request regardless of the physical position of the vehicle and whether or not the S/R requests are the same.

Next, we discuss how to get the probabilities of different classes of customers. We assume arrival processes of these classes of customers are Poisson and denote the arrival rate as λ_i, $i = 1, ..., 16$. According to the property of the Poisson processes, the total arrival rate λ is:

$$\lambda = \sum_{i=1}^{16} \lambda_i. \tag{6.1}$$

In real applications, we only know the value of throughput of storage request λ_s and throughput of retrieval request λ_r. Obviously,

$$\lambda = \lambda_s + \lambda_r. \tag{6.2}$$

According to this equation, the probabilities of storage request and retrieval request are:

$$P_s = \frac{\lambda_s}{\lambda}, P_r = \frac{\lambda_r}{\lambda}. \tag{6.3}$$

We denote the number of tiers in AVS/RS is T. The probabilities of the 16 classes of customers are shown in Table 6.1.

6.2.2. Service network

There are 8 service stages in the AVS/RS: S_j, $j = 1...8$. The first server S_1 includes two operations. At first, the vehicle travels from the lift position to the designated S/R position. The second operation is the drop-off time for a storage transaction or the pick-up time for a retrieval transaction. S_2 denotes the travel time of a vehicle from the position where the previous transaction was completed to the lift position. S_3 is the travel time of the lift from the previous tier to the designated tier. S_4 includes

Table 6.1. Probabilities of 16 classes of customers

Customer Class (C_i)		P_i
C_1	(S,G,I)	$P_1 = P_s \frac{1}{T} \frac{1}{T+2}$
C_2	(S,G,O)	$P_2 = P_s \frac{1}{T} \frac{1}{T+2}$
C_3	(S,G,G)	$P_3 = P_s \frac{1}{T} \frac{1}{T+2}$
C_4	(S,G,Ot)	$P_4 = P_s \frac{1}{T} \frac{T-1}{T+2}$
C_5	(S,Ot,I)	$P_5 = P_s \frac{T-1}{T} \frac{1}{T+2}$
C_6	(S,Ot,O)	$P_6 = P_s \frac{T-1}{T} \frac{1}{T+2}$
C_7	(S,Ot,G)	$P_7 = P_s \frac{T-1}{T} \frac{1}{T+2}$
C_8	(S,Ot,Ot)	$P_8 = P_s \frac{T-1}{T} \frac{T-1}{T+2}$
C_9	(R,G,I)	$P_9 = P_r \frac{1}{T} \frac{1}{T+2}$
C_{10}	(R,G,O)	$P_{10} = P_r \frac{1}{T} \frac{1}{T+2}$
C_{11}	(R,G,G)	$P_{11} = P_r \frac{1}{T} \frac{1}{T+2}$
C_{12}	(R,G,Ot)	$P_{12} = P_r \frac{1}{T} \frac{1}{T+2}$
C_{13}	(R,Ot,I)	$P_{13} = P_r \frac{T-1}{T} \frac{1}{T+2}$
C_{14}	(R,Ot,O)	$P_{14} = P_r \frac{T-1}{T} \frac{1}{T+2}$
C_{15}	(R,Ot,G)	$P_{15} = P_r \frac{T-1}{T} \frac{1}{T+2}$
C_{16}	(R,Ot,Ot)	$P_{16} = P_r \frac{T-1}{T} \frac{T-1}{T+2}$

two operations. The first part is the pick-up time for a storage request and the second part is the travel time between the input point and the lift position. Similar to S_4, S_5 also includes two operations. The first part is the drop-off time for a retrieval request and the second part is the travel time between the output point and the lift position. S_6 denotes the travel time of a paired vehicle to pick up the load at the input point. So S_6 only includes the travel time between the input point and the lift position. Similar to S_6, S_7 includes the travel time between the output point and the lift position. S_8 denotes travel time between the I/O point, which is selected only when the available vehicle is at the output point for the storage request.

As discussed earlier, there are 16 classes of customers in the AVS/RS. Each class of customers has its own route. For instance, C_8 **(S,Ot,Ot)** denotes that the storage request is on a tier other than the ground floor and the available vehicle is also on a tier other than the ground floor. To complete this transaction, the vehicle travels to the lift position and is taken to the ground floor by the lift. Then the vehicle travels

from the lift position to the input point to pick up the load. After that, the vehicle travels back to the lift position and is taken by the lift to the designated tier. Finally, the vehicle travels to the storage place and drops off the load. The route of C_8 can be described as $S_2 \rightarrow S_3 \rightarrow S_6 \rightarrow S_4 \rightarrow S_3 \rightarrow S_1$.

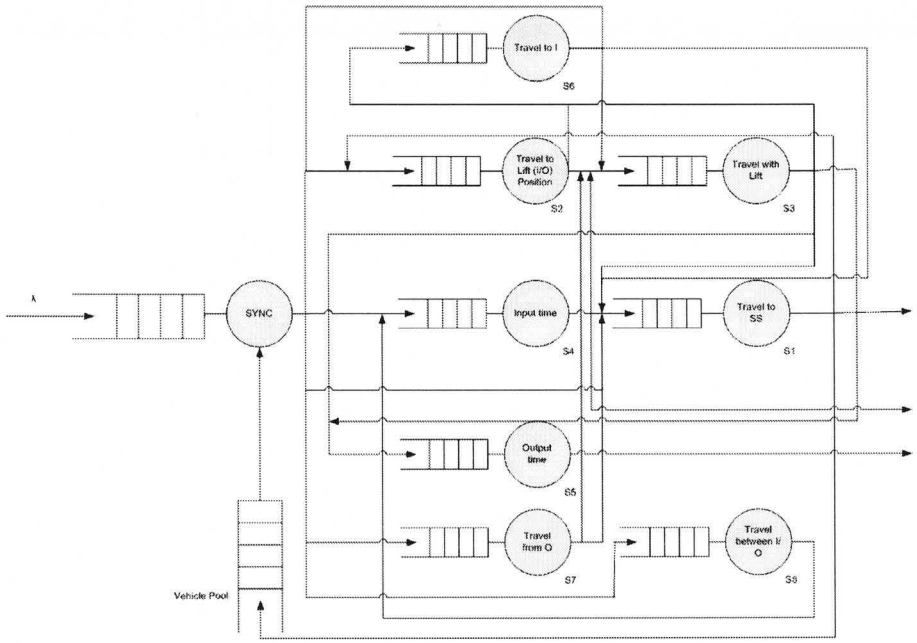

Figure 6.1. SOQN model of AVS/RS

Figure 6.1 shows the SOQN model of the AVS/RS and Table 6.2 shows the routes of the 16 classes of customers.

In this SOQN model, we noticed that only the queue in front of S_3 is a physical queue. Each service stage except S_3 has parallel servers. The number of servers in S_3 is the number of lifts.

6.2.3. Visit ratios

Since there are multiple classes of customers in this model, it is necessary to calculate the visit ratio at each node. First, the routing probabilities between each pair of

Table 6.2. Sequence of servers visited by 16 customer classes

Route/Index	1	2	3	4	5	6
(S,G,I)	S_4	S_1				
(S,G,O)	S_8	S_4	S_1			
(S,G,G)	S_2	S_6	S_4	S_1		
(S,G,Ot)	S_2	S_3	S_6	S_4	S_1	
(S,Ot,I)	S_4	S_3	S_1			
(S,Ot,O)	S_8	S_4	S_3	S_1		
(S,Ot,G)	S_2	S_6	S_4	S_3	S_1	
(S,Ot,Ot)	S_2	S_3	S_6	S_4	S_3	S_1
(R,G,I)	S_6	S_1	S_2	S_5		
(R,G,O)	S_7	S_1	S_2	S_5		
(R,G,G)	S_1	S_2	S_5			
(R,G,Ot)	S_2	S_3	S_1	S_2	S_5	
(R,Ot,I)	S_6	S_3	S_1	S_2	S_3	S_5
(R,Ot,O)	S_7	S_3	S_1	S_2	S_3	S_5
(R,Ot,G)	S_2	S_3	S_1	S_2	S_3	S_5
(R,Ot,Ot)	S_2	S_3	S_1	S_2	S_3	S_5

nodes are calculated by the following equation:

$$P_{ij} = \frac{P_{pair} + P_{first}P_{last}}{P_{out}}, \tag{6.4}$$

where P_{ij} is the routing probability, P_{pair} is the probability a customer leaves from node i to node j, P_{first} is the probability that the first node of the route is node j, P_{last} is the probability that the last node of the route is node i, and P_{out} is the probability that a customer leaves from node i. For example, P_{12} is the routing probability from

S_1 to S_2, and it can be calculated as:

$$P_{pair} = P_{(R,G,I)} + P_{(R,G,O)} + P_{(R,G,G)} + P_{(R,G,Ot)} + P_{(R,Ot,I)} + P_{(R,Ot,O)} + P_{(R,Ot,G)} + P_{(R,Ot,Ot)},$$

$$P_{first} = P_{(S,G,G)} + P_{(S,G,Ot)} + P_{(S,Ot,G)} + P_{(S,Ot,Ot)} + P_{(R,G,Ot)} + P_{(R,Ot,G)} + P_{(R,Ot,Ot)},$$

$$\begin{aligned} P_{last} &= P_{(S,G,I)} + P_{(S,G,O)} + P_{(S,G,G)} + P_{(S,G,Ot)} + P_{(S,Ot,I)} + P_{(S,Ot,O)} + P_{(S,Ot,G)} + P_{(S,Ot,Ot)} \\ &= P_s, \end{aligned}$$

$$\begin{aligned} P_{out} &= P_{(S,G,I)} + P_{(S,G,O)} + P_{(S,G,G)} + P_{(S,G,Ot)} + P_{(S,Ot,I)} + P_{(S,Ot,O)} + P_{(S,Ot,G)} + P_{(S,Ot,Ot)} \\ &\quad + P_{(R,G,I)} + P_{(R,G,O)} + P_{(R,G,G)} + P_{(R,G,Ot)} + P_{(R,Ot,I)} + P_{(R,Ot,O)} + P_{(R,Ot,G)} + P_{(R,Ot,Ot)} \\ &= P_s + P_r = 1, \end{aligned}$$

and

$$P_{12} = \frac{P_{pair} + P_{first}P_{last}}{P_{out}} = 0.913.$$

Table 6.3 shows the routing probability of each pair of nodes.

Table 6.3. Routing probabilities

P_{ij}	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_1	0.0037	0.913	0	0.0156	0	0.026	0.026	0.0156
S_2	0	0	0.906	0	0.0641	0.0299	0	0
S_3	0.5385	0	0	0	0.3148	0.1468	0	0
S_4	0.1429	0	0.8571	0	0	0	0	0
S_5	0.0099	0.7679	0	0.0416	0	0.0695	0.0695	0.0416
S_6	0.0275	0	0.1649	0.8076	0	0	0	0
S_7	0.1429	0	0.8571	0	0	0	0	0
S_8	0	0	0	1	0	0	0	0

Routing probabilities should satisfy the following equation:

$$\sum_{j=1}^{8} P_{ij} = 1, \text{ for } i = 1, 2, \cdots, 8. \tag{6.5}$$

The visit ratio of node i, e_i, can be calculated by:

$$e_i = \sum_{j=1}^{8} e_j P_{ji}, \text{ for } i = 1, 2, \cdots, 8. \tag{6.6}$$

We can set e_1 equal 1 to get 8 independent equations for visit ratios. Table 6.4 shows these visit ratios.

Table 6.4. Visit ratios

Servers	e_i
S_1	1
S_2	1.393
S_3	1.7024
S_4	0.3748
S_5	0.6252
S_6	0.361
S_7	0.0695
S_8	0.0416

6.2.4. A simplified case

In an implementation of the AVS/RS, the warehouse is divided into several zones. Each zone has its own rack system, I/O points, vehicles and lifts. Modeling each zone can be simplified by assuming the I/O points and the lift position on the ground floor are in the same position: the left lower corner of the rack system. The rack system is shown in the right side of figure 3.8. Thus, the number of customer classes is reduced to 8, and the number of service stages is reduced to 4 in each zone.

Figure 6.2 shows the SOQN model of each zone. We can calculate routing probabilities and visit ratios in the same way (shown in Tables 6.5 and 6.6).

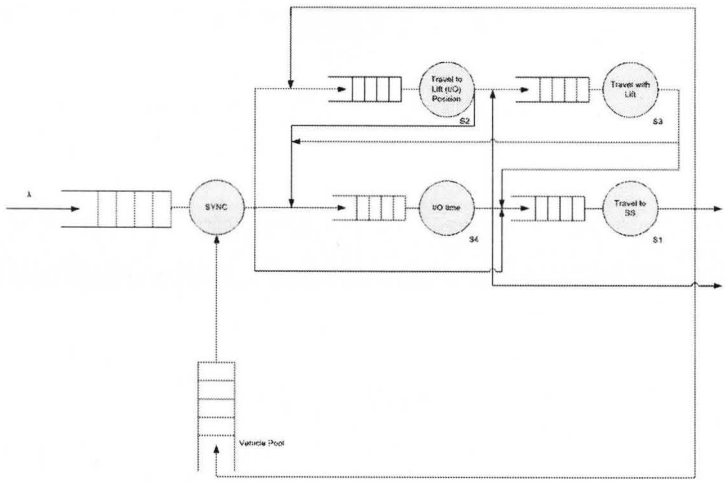

Figure 6.2. The SOQN model of each zone

Table 6.5. Sequence of servers visited by customers of the simplified case

Route/Index	1	2	3	4	5	6
(S,G,I)	S_4	S_1				
(S,G,G)	S_2	S_4	S_1			
(S,G,Ot)	S_2	S_3	S_4	S_1		
(S,Ot,I)	S_4	S_3	S_1			
(S,Ot,G)	S_2	S_4	S_3	S_1		
(S,Ot,Ot)	S_2	S_3	S_6	S_4	S_3	S_1
(R,G,G)	S_1	S_2	S_4			
(R,G,Ot)	S_2	S_3	S_1	S_2	S_4	
(R,Ot,G)	S_2	S_3	S_1	S_2	S_3	S_4
(R,Ot,Ot)	S_2	S_3	S_1	S_2	S_3	S_4

Table 6.6. Routing probabilities and visit ratios of the simplified case

P_{ij}	S_1	S_2	S_3	S_4	e_i
S_1	0.004782	0.977657	0	0.017561	1
S_2	0	0	0.920528	0.083378	1.5604
S_3	0.533333	0	0	0.466667	1.7542
S_4	0.061521	0.58972	0.321268	0.029291	0.9939

6.3. Discussion of synchronization process policies

6.3.1. Physical synchronization process

In Section 6, we gave a simplified method to estimate the travel time of the vehicle before it arrives at the position where the request is generated. However, this method

may lead to a huge error in evaluating the performance of the system. Firstly, the relative error of travel time with lift is larger when the numbers of tiers become larger. Secondly, this method cannot deal with the situation when the S/R throughput of each tier is different. Higher throughput at a certain tier increases the probability that the available vehicle is at that tier instead of the other tiers. Simply assuming that the probabilities of the available vehicle at any tier are the same will lead to an inaccurate result in this case.

It is necessary to make classes of customers independent from available vehicle locations. The physical synchronization process policy is a possible solution to solve this problem. Under this policy, all vehicles need to travel back to the I/O point after the S/R transaction is completed.

We can remodel the AVS/RS as the SOQN under this synchronization policy. The synchronization process occurs at the I/O point. Now, S/R requests at different tiers can be treated as different classes of customers. Similar to the simplified model described in Section 6, we assume the whole rack system is divided into several zones. In each zone, the I/O points and the lift position on the ground floor are in the same position. Each zone has its own vehicles. All vehicles need to come back to I/O point after the transaction is completed.

The customers of this SOQN can be redefined as follows. A class of customers is defined by the transaction type and the destination/original tier. Similar to the SOQN model in Section 6, there are two types of transactions: storage request (**S**) and retrieval request (**R**). As before, we separate the S/R requests on the ground floor (**G**) and on other tiers (Ot_i). Ot_i means ith any tier other than the ground floor. For example, (**S**, Ot_2) is a class of storage requests on the 2nd tier. According to this new definition, there are $2T$ classes of customers, where T is the number of

tiers. Now, we can handle the situation where the throughput of S/R transactions are different.

There are two kinds of service stages in this SOQN model. The first represents the travel time with the lift. This stage is a multi-server stage and the number of parallel servers is the number of lifts. The second service stage is the travel time on different tiers. Because all vehicles need to come back to I/O points, the expected service time is two times the expected travel time on the tier. Similar to the SOQN model in the previous section, there is no physical queue in front of these stages. We need to assign enough parallel servers to eliminate queues in these stages. In this SOQN model, the completion of a S/R request does not mean the customer leaves the network. The moment when the vehicle travels back to I/O points is when the customer is assumed to leave the network.

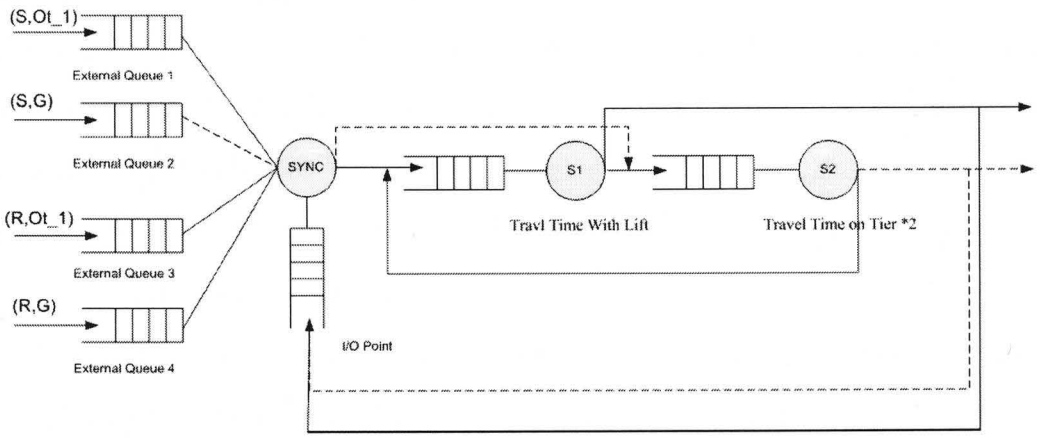

Figure 6.3. The physical synchronization process

Figure 6.3 shows how the physical synchronization works. In order to keep this SOQN model simple and clear, we assume the number of tiers T is 2. Hence, we have 4 classes of customers: (**S**, Ot_1), (**S**, **G**), (**R**, Ot_1) and (**R**, **G**). Additionally, we assume the distributions of vehicle travel times on all tiers are identical.

Although the SOQN model based on the physical synchronization policy can improve the accuracy of our method, the AVS/RS with a physical synchronization process is not as efficient as the AVS/RS with a virtual synchronization process. The reason is that the vehicles need to travel back to the I/O points after the transaction is completed, which occupies lifts and wastes time. In real applications, the virtual synchronization policy is a reasonable choice. Hence, improving the accuracy of our method in estimating the AVS/RS with virtual synchronization process is a possible direction of future research.

6.3.2. Virtual synchronization process

In Section 6 we discussed the synchronization process in the SOQN model of the AVS/RS. The synchronization process in the SOQN means an available vehicle is assigned to an S/R request no matter where the physical position of the vehicle is. This process is a virtual synchronization process because the S/R request cannot be processed until the attached vehicle travels to the position where the request originates. For a storage request, the pallet needs to wait for the available vehicle to travel to the I/O point. For a retrieval request, the pallet needs to wait for the available vehicle to travel to the storage position.

The corresponding SOQN model based on this synchronization process has difficulty in estimating the distribution of travel time of the vehicle before arriving the request generating position. According to the discussion in Section 6, a particular class of customers can be defined by the S/R request, the destination/original position and the available vehicle position. For example, for a storage request of a pallet at the I/O position, the available vehicle could be at any storage position in the rack system. The expected travel time could be divided into two parts: the travel time from the current position to the lift position at the particular tier and the travel time

with lift from that tier to the ground floor. The vehicle travel time can be estimated by the method we described in Section 3. However, it is hard to estimate the travel time with lift for the reason that it is impossible to know which tier the available vehicle is at before the synchronization process occurs. This synchronization process leads to a contradiction in modeling the AVS/RS by the SOQN: the class of customers depends on the location of the available vehicle. However, resources in the SOQN are identical, and the distribution of service time should be independent from the resource.

An alternate method to estimate the travel time with lift is assuming the probabilities that the vehicle at any tier are same. The lift travel time is then the average of travel times to different tiers.

6.4. Numerical experiments

6.4.1. Physical synchronization process

As we discussed in the previous section, the physical synchronization process requires all vehicles to travel back to I/O points after completing their S/R transactions. In the corresponding SOQN model, a customer represents the entire set of vehicle transactions, rather than that for a pallet.

In order to compare the tier to tier AVS/RS with the AVS/RS with tier-captive vehicles, we conduct experiments on an AVS/RS having the same parameters as we discussed in Section 3. There are 7 tiers in the rack system. Hence, there are 7 storage request classes and 7 retrieval request classes. The average travel time of a lift is 0.46 minutes. Additionally, the average travel time of a vehicle in a tier is 1.753 minutes. As before, we assume the location of I/O points overlaps with the location of lifts on the ground floor. Under this assumption, constant travel times between the lifts and I/O points are omitted.

In the first set of experiments, we discuss the performance of this AVS/RS under the normal throughput requirement, 250 pallets per hour for S/R transactions. Table 6.7 shows the queue lengths of the aggregated class of customers and each storage class and retrieval class of customers. Here, we assume that the throughput of each tier of S/R requests is identical.

Table 6.7. AVS/RS with physical synchronization process $\lambda_s = \lambda_r = 250\ pallets/hr$

Aggregated Class		ith Storage Request		ith Retrieval Request	
L_{eq}	L_{lift}	L_{eq}	L_{lift}	L_{eq}	L_{lift}
V = 2, L = 5					
5.12	1.55	0.366	0.129	0.366	0.129
V = 4, L = 3					
20.3	2.74	1.45	0.230	1.45	0.230
V = 4, L = 4					
6.78	3.21	0.48	0.280	0.484	0.280

The second set of experiments shows how to adjust the number of lifts and vehicles in the AVS/RS to meet different throughput requirements. In table 6.8, there are three different throughput requirements, which represent high, normal and low throughput scenarios respectively. As in the first set of experiments, the throughput of each tier of S/R requests is identical.

Table 6.8. AVS/RS with physical synchronization process, varied throughput

Aggregated Class		ith Storage Request		ith Retrieval Request	
L_{eq}	L_{lift}	L_{eq}	L_{lift}	L_{eq}	L_{lift}
$\lambda_s = \lambda_r = 500 pallets/hr$ V = 6, L = 8					
2.35	3.78	0.168	0.315	0.168	0.315
$\lambda_s = \lambda_r = 250 pallets/hr$ V = 2, L = 5					
5.12	1.55	0.366	0.129	0.366	0.129
$\lambda_s = \lambda_r = 125 pallets/hr$ V = 2, L = 3					
1.49	1.18	0.167	0.099	0.167	0.099

Again, table 6.8 indicates that the AVS/RS is very flexible. By adjusting the number of vehicles and lifts, we can keep the AVS/RS at a stable performance level.

We generalize the AVS/RS in first two sets of experiments a further. In the previous experiments, we assumed the throughput of each tier of S/R requests is identical. The throughput of each tier could be different from other tiers, however. Here we assume the throughput of S/R transactions on the ground floor is different from those in other tiers. The overall throughput is still 250 pallets per hour. In this set of experiments, the throughput on the ground floor is 125 pallets per hour, which takes 50% of the entire throughput. The throughput on each tier other than the ground floor is identical, which is about 125/6 pallets per hour.

Table 6.9. AVS/RS with physical synchronization process, uneven throughtput

Aggregated Class		Ground Floor S/R Request		ith S/R Request	
L_{eq}	L_{lift}	L_{eq}	L_{lift}	L_{eq}	L_{lift}
V = 4, L = 4					
0.040	0.842	0.012	N/A	0.001	0.070
V = 4, L = 2					
18.31	3.56	4.92	N/A	0.705	0.300
V = 8, L = 2					
3.12	5.26	0.84	N/A	0.120	0.438

The last set of experiments for the AVS/RS with physical synchronization process is about the general distribution. Similar to the experiments conducted in Chapter 5, the inter-arrival times and service times follow high and low variance distributions, respectively. Table 6.10 shows the result of this set of experiments.

6.4.2. Virtual synchronization process

From the comparison between the physical synchronization process and the virtual synchronization process, it is obvious to conclude that the performance of AVS/RS with a virtual synchronization process is better than the performance of the AVS/RS with a physical synchronization process. In the AVS/RS with a physical synchronization process, the synchronization of the pallet and the vehicle always occurs at the I/O point on the ground floor. The vehicle needs to travel back to the I/O point after

Table 6.10. AVS/RS with physical synchronization process, uneven throughtput

Aggregated Class		ith Storage Request		ith Retrieval Request	
L_{eq}	L_{lift}	L_{eq}	L_{lift}	L_{eq}	L_{lift}
$C_{Xa}^2 = 0.5$, $C_{Xl}^2 = 2$					
V = 4, L = 4					
12.6	3.53	0.901	0.294	0.901	0.294
V = 6, L = 4					
3.51	4.32	0.251	0.360	0.251	0.360
$C_{Xa}^2 = 2$, $C_{Xl}^2 = 0.5$					
V = 4, L = 4					
7.37	3.12	0.536	0.260	0.536	0.260
V = 6, L = 4					
4.38	4.09	0.313	0.340	0.313	0.340

it completes the S/R transaction. On the other hand, vehicles in the AVS/RS with virtual synchronization process are paired with waiting pallets immediately after the completion of the previous transaction. In other words, the performance of AVS/RS with physical synchronization process is always the lower bound of the performance of AVS/RS with a virtual synchronization process.

In this part of experiments, we still utilize the same system as in the previous subsection of the physical synchronization process. Different from the AVS/RS with physical synchronization process, customer classes in the AVS/RS with virtual synchronization process are independent of the number of tiers in the system. Table 6.11 shows probabilities of 12 different classes of customers in the AVS/RS with a virtual synchronization process.

Table 6.11. Probabilities of 12 different classes

Index	Class	Prob	Index	Class	Prob
C1	(S,G,I)	1/28	C7	(R,G,I)	1/28
C2	(S,G,G)	1/196	C8	(R,G,G)	1/196
C3	(S,G,Ot)	6/196	C9	(R,G,Ot)	6/196
C4	(S,Ot,I)	6/28	C10	(R,Ot,I)	6/28
C5	(S,Ot,G)	6/196	C11	(R,Ot,G)	6/196
C6	(S,Ot,Ot)	36/196	C12	(R,Ot,Ot)	36/196

In the first set of experiments, we discuss the performance of this AVS/RS under the normal throughput requirement, 250 pallets per hour for S/R transactions. Table 6.12 to table 6.14 show the queue lengths of the aggregated class of customers and each storage and retrieval class of customers for different sets of vehicles and lifts.

Table 6.12. AVS/RS with virtual synchronization process $\lambda_s = \lambda_r =$ 250 *pallets/hr* (1)

V = 4, L = 4						
Aggregated Class $L_{eq} = 4.41$, $L_{lift} = 2.89$						
Index	C1	C2	C3	C4	C5	C6
L_{eq}	0.157	0.023	0.135	0.944	0.135	0.809
L_{lift}	N/A	N/A	0.081	0.569	0.081	0.488
Index	C7	C8	C9	C10	C11	C12
L_{eq}	0.157	0.023	0.135	0.944	0.135	0.809
L_{lift}	N/A	N/A	0.081	0.794	0.113	0.681

Table 6.13. AVS/RS with virtual synchronization process $\lambda_s = \lambda_r =$ 250 *pallets/hr* (2)

V = 3, L = 4						
Aggregated Class $L_{eq} = 5.24$, $L_{lift} = 2.31$						
Index	C1	C2	C3	C4	C5	C6
L_{eq}	0.187	0.027	0.161	1.123	0.161	0.963
L_{lift}	N/A	N/A	0.062	0.432	0.062	0.371
Index	C7	C8	C9	C10	C11	C12
L_{eq}	0.187	0.027	0.161	1.123	0.161	0.963
L_{lift}	N/A	N/A	0.062	0.657	0.094	0.563

Table 6.14. AVS/RS with virtual synchronization process $\lambda_s = \lambda_r =$ 250 *pallets/hr* (3)

V = 2, L = 5						
Aggregated Class $L_{eq} = 1.58$, $L_{lift} = 1.21$						
Index	C1	C2	C3	C4	C5	C6
L_{eq}	0.056	0.008	0.048	0.338	0.048	0.290
L_{lift}	N/A	N/A	0.030	0.21	0.03	0.179
Index	C7	C8	C9	C10	C11	C12
L_{eq}	0.056	0.008	0.048	0.338	0.048	0.290
L_{lift}	N/A	N/A	0.030	0.368	0.052	0.314

We notice that L_{lift}s of retrieval requests on tiers other than the ground floor are longer than those of storage requests. This demonstrates that these retrieval requests are time consuming because of the additional service step utilizing the lift .

Additionally, the performance comparison between the aggregated class of AVS/RS with virtual synchronization process and the aggregated class of AVS/RS with physical synchronization process indicates that the performance of AVS/RS with virtual synchronization process is always better.

The second set of experiments shows how to adjust the number of lifts and vehicles in the AVS/RS to meet different throughput requirements. Tables 6.15, 6.14 and 6.16 show different sets of vehicles and lifts for normal, high and low throughput requirements respectively.

Table 6.15. AVS/RS with virtual synchronization process $\lambda_s = \lambda_r = 500$ *pallets/hr*

V = 6, L = 8						
Aggregated Class $L_{eq} = 0.833$, $L_{lift} = 2.694$						
Index	C1	C2	C3	C4	C5	C6
L_{eq}	0.030	0.004	0.026	0.179	0.026	0.153
L_{lift}	N/A	N/A	0.078	0.544	0.078	0.467
Index	C7	C8	C9	C10	C11	C12
L_{eq}	0.030	0.004	0.026	0.179	0.026	0.153
L_{lift}	N/A	N/A	0.078	0.725	0.104	0.621

Table 6.16. AVS/RS with virtual synchronization process $\lambda_s = \lambda_r = 125$ *pallets/hr*

V = 2, L = 3						
Aggregated Class $L_{eq} = 0.641$, $L_{lift} = 0.394$						
Index	C1	C2	C3	C4	C5	C6
L_{eq}	0.023	0.003	0.020	0.137	0.020	0.118
L_{lift}	N/A	N/A	0.078	0.544	0.078	0.467
Index	C7	C8	C9	C10	C11	C12
L_{eq}	0.023	0.003	0.020	0.137	0.020	0.118
L_{lift}	N/A	N/A	0.023	0.286	0.041	0.246

In the third set of experiments, we present the AVS/RS with uneven S/R throughput requirements on the ground floor and other tiers and generally distributed inter-arrival and service times. As before, we assume the S/R throughput on the ground floor represents 50% of the total throughput requirement. The total throughput is 250 pallets per hour, and the throughput on the ground floor is 125 pallets per hour. Table 6.17 shows the probabilities of 12 different classes of customers in the AVS/RS with virtual synchronization process under this assumption.

Table 6.17. Probabilities of 12 different classes, uneven throughput

Index	Class	Prob	Index	Class	Prob
C1	(S,G,I)	1/8	C7	(R,G,I)	1/8
C2	(S,G,G)	1/16	C8	(R,G,G)	1/16
C3	(S,G,Ot)	1/16	C9	(R,G,Ot)	1/16
C4	(S,Ot,I)	1/8	C10	(R,Ot,I)	1/8
C5	(S,Ot,G)	1/16	C11	(R,Ot,G)	1/16
C6	(S,Ot,Ot)	1/16	C12	(R,Ot,Ot)	1/16

Table 6.18 presents the result of this AVS/RS with exponentially distributed inter-arrival and service times. In table 6.19, the distribution of the inter-arrival times has a low variance and the distribution of service times has a hight variance. On the other hand, the inter-arrival times have higher variance than the service times in table 6.20.

Table 6.18. AVS/RS with virtual synchronization process, uneven throughput (1)

$C_{Xa}^2 = 1$, $C_{Xl}^2 = 1$						
V = 8, L = 2						
Aggregated Class $L_{eq} = 1.05$, $L_{lift} = 2.95$						
Index	C1	C2	C3	C4	C5	C6
L_{eq}	0.131	0.066	0.066	0.131	0.066	0.066
L_{lift}	N/A	N/A	0.261	0.521	0.261	0.261
Index	C7	C8	C9	C10	C11	C12
L_{eq}	0.131	0.066	0.066	0.131	0.066	0.066
L_{lift}	N/A	N/A	0.261	0.692	0.346	0.346

Table 6.19. AVS/RS with virtual synchronization process, uneven throughput (2)

$C_{Xa}^2 = 0.5$, $C_{Xl}^2 = 2$						
V = 6, L = 4						
Aggregated Class $L_{eq} = 1.57$, $L_{lift} = 1.91$						
Index	C1	C2	C3	C4	C5	C6
L_{eq}	0.197	0.098	0.098	0.197	0.098	0.098
L_{lift}	N/A	N/A	0.154	0.308	0.154	0.154
Index	C7	C8	C9	C10	C11	C12
L_{eq}	0.197	0.098	0.098	0.197	0.098	0.098
L_{lift}	N/A	N/A	0.154	0.493	0.247	0.247

Table 6.20. AVS/RS with virtual synchronization process, uneven throughput (3)

$C_{Xa}^2 = 2$, $C_{Xl}^2 = 0.5$						
V = 6, L = 4						
Aggregated Class $L_{eq} = 0.682$, $L_{lift} = 1.32$						
Index	C1	C2	C3	C4	C5	C6
L_{eq}	0.085	0.043	0.043	0.085	0.043	0.043
L_{lift}	N/A	N/A	0.105	0.211	0.105	0.105
Index	C7	C8	C9	C10	C11	C12
L_{eq}	0.085	0.043	0.043	0.085	0.043	0.043
L_{lift}	N/A	N/A	0.105	0.324	0.162	0.162

Again, we demonstrate that the performance of the AVS/RS with a physical synchronization process is the lower bound of the performance of the AVS/RS with a virtual synchronization process even when the throughput is not evenly distributed.

6.5. Conclusions

In this chapter, we discussed how to apply SOQN models in performance evaluation of AVS/RS with tier-to-tier vehicles. First, we discussed how to model an AVS/RS with tier-to-tier vehicles as an SOQN model in Section 6. There are three necessary elements in an SOQN model: customer, service node, service path. S/R requests occurring in different locations of the AVS/RS are modeled as customers in the SOQN model. Different travel times of the AVS/RS are modeled as service times

of service nodes in the SOQN model. The service path of a certain class of customers is the execution sequence of a certain class of S/R requests.

Second, we compared AVS/RSs with two different synchronization processes. The physical synchronization process always requires the vehicles to travel back to the I/O points after it completes each S/R transaction. In the AVS/RS with virtual synchronization process, the vehicle stays at the position where the last transaction is completed and is synchronized with the next S/R request virtually. We demonstrate that the performance of AVS/RS with a physical synchronization process is the lower bound of the performance of AVS/RS with a virtual synchronization process. This conclusion is verified in a set of numerical experiments.

CHATPER 7

SUMMARY

7.1. Conclusions

In this thesis, we discussed how to evaluate the performance of two automated warehouse design technologies - AS/RS and AVS/RS. AS/RS is a popular and mature technology that has been researched well in the literature. We selected some classic papers of AS/RS to present some well-established analytical and simulation models in Chapter 1. On the other hand, AVS/RS, our research concentration, is a new technology that was introduced less than two decades ago. There are two types of possible AVS/RS technologies: AVS/RS with tier-captive vehicles and AVS/RS with tier-to-tier vehicles. There are not well-established models in the literature to analyze the performance of this technology. We presented some what we believe is pioneering work in this thesis.

The AS/RS and AVS/RS have both been used in automated warehouses to improve their operational performance metrics. Our goal in this thesis is to analyze complex stochastic systems such as automated warehouses. There are many ways to analyze stochastic systems. We employ a simple, but powerful mathematical tool, queueing network model. We conducted a literature review of classic queueing network models and corresponding algorithms in Chapter 1. There are three important types of queueing network models, OQN, CQN and SOQN. OQN and CQN have been studied thoroughly. SOQN has not been as widely studied There are not many established algorithms for SOQN in the literature. We adopted the MGM to solve it.

We used OQN to analyze AS/RS and AVS/RS wither tier-captive vehicles. For AVS/RS with tier-to-tier vehicles, we used SOQN to evaluate its performance.

In Chapter 3, we modeled both the AS/RS and the AVS/RS with tier-captive vehicles as OQN models. We adopted MPA, a tool to evaluate performance of OQN models to estimate performance measures of these two technologies.

In Chapter 4, we discussed how to use MGM to solve SOQN models. It is always a good idea to start with simple cases. Hence, we assume all SOQN models in this chapter have exponentially distributed inter-arrival and service times. However, exponential distributions have limiting assumptions, e.g. memoryless property, which prevents its applications in the real world. We extended our algorithms to general distribution cases in Chapter 5. Although algorithms for general distributions are much more complicated, we decomposed them into simpler sub-models and applied algorithms we obtained in Chapter 4 to solve them.

In Chapter 6, we applied SOQN models and corresponding algorithms from Chapter 5 to estimate performance measures of an AVS/RS with tier-to-tier vehicles.

7.2. Future research plan

We presented efficient algorithms to evaluate performance of SOQN models in our thesis. However, the ability of these algorithms to handle large-scale SOQN models is limited. One possible direction of further research could be to develop algorithms that can solve larger problems effectively.

Secondly, as mentioned in Chapter 1, there are few papers about performance evaluation of AVS/RS in literature. Warehouse designers lack good software tools to evaluate candidate designs easily. Hence, we plan to develop a user-friendly software package and embed our approximate algorithms in it. The goal of our further

research is to integrated efficient algorithms of AS/RS, AVS/RS with tier-captive vehicles and AVS/RS with tier-to-tier vehicles we have into a unified software package. This application will be a useful tool to support automatic warehouse designers to evaluate their designs effectively and efficiently. Figure 7.1 shows a demo version of this software package.

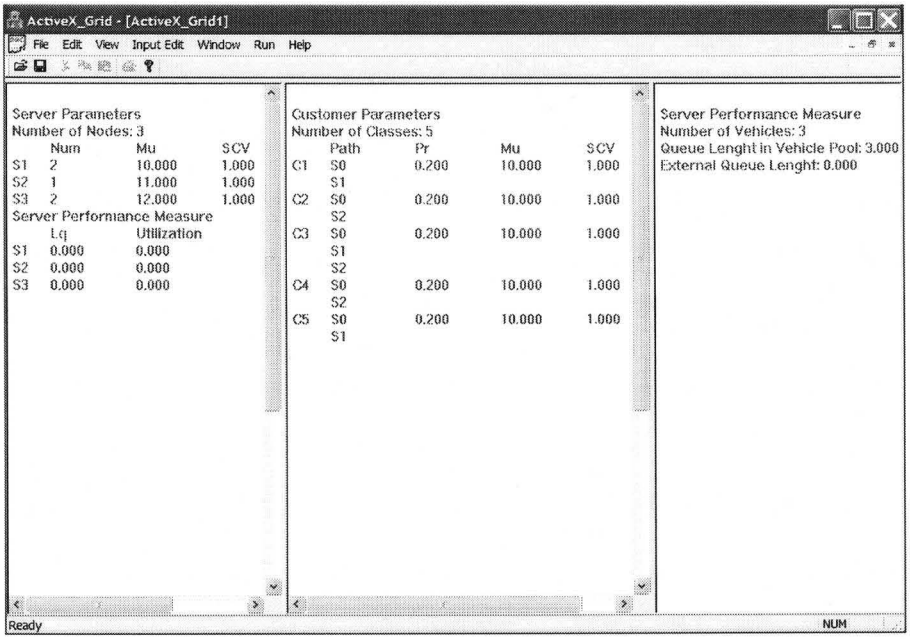

Figure 7.1. A demo software package for automatic warehouse design

REFERENCES

N. Akar, N. Oguz, and K. Sohraby. Matrix-geometric solutions of m/g/1-type markov chains: a unifying generalized state-space approach. *IEEE Journal on Selected Areas in Communications*, 16:626–639, 1998.

J. Ashayeri, R. Heuts, M.W.T.Valkenburg, H. Veraart, and M. Wilhelm. A geometrical approach to computing expected cycle times for zone-based storage layouts in as/rs. *Int. J. Prod. Res.*, 40:4467–4483, 2002.

B. Avi-Itzhak and D. Heyman. Approximate queuing models for multiprogramming computer systems. *Operations Research*, 21:1212–1230, 1973.

Y. Bard. Some extensions to multiclass queueing network analysis. In *4th Int. Symp. on Modelling and Performance Evaluation of Cumpter Systems*, 1979.

F. Baskett, K. Chandy, R. Muntz, and F. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of ACM*, 22:248–260, 1975.

B. Baynat and Y. Dallery. A product-form approximation method for general closed queueing networks with several classes of customers. *Performance Evaluation*, 24: 165–188, 1996.

R. Bellman. *Introduction to Matrix Analysis.* McGraw-Hill, 1960.

A. Benamar, Z. Sari, and N. Ghouali. Performance analysis for multi-aisle automated storage/retrieval systems using visual petri net developer. In *Proceedings 2003 IEEE International Symposium on Computatioal Intelligence in Robotics and Automation*, 2003.

J. Berg and A. Gademann. Simulation study of an automated storage/retrieval system. *International Journal of Production Research*, 38:1339–1356, 2000.

J. Bird. *Electrical Circuit Theory and Technology.* Newnes, 2007.

G. Bitran and S. Dasu. A review of open queueing network of manufacturing systems. *Queueing Systems*, 12:95–133, 1992.

G. Bitran and D. Tirupati. Multiproduct queueing networks with deterministic routing: decomposition approach and the notion of interference. *Management Science*, 35:851–878, 1988.

G. Bolch, G. S., H. Meer, and K. Trivedi. *Queueing Networks and Markov Chains.* Wiley-interscience publication, 1998.

Y. Bozer and J. White. Travel time models for automated stroage/retrieval systems. *IIE Transactions*, 16:329–338, 1984.

A. Brandwajn. Fast approximate solution of multiprogramming models. *ACM Performance Evaluation Review*, 11:141–149, 1982.

S. Bruell and G. Balbo. Computational algorithms for closed queueing networks. *Operating and Programming Systems Series*, 1980.

S. Bruell, G. Balbo, and P. Afshari. Mean value analysis of mixed, multiple class bcmp networks with load dependent service stations. *Performance Evaluation*, 4: 241–260, 1984.

R. Buitenhek. *Performance evaluation of dual resource manufacturing systems.* PhD thesis, Univeristy of Twente, The Netherlands, 1998.

R. Buitenhek, G. Houtum, and H. Zijm. Amva-based solution procedures for open queueing networks with population constraints. *Annals of Operations Research*, 93: 15–40, 2000.

J. Buzacott. On the optimal control of input to a job shop. In *ORSA-TIMS meeting*, 1974.

J. Buzacott and J. Shanthikumar. Models for understanding flexible manufacturing systems. *AIIE Transactions*, 12:339–350, 1980.

J. Buzacott and J. Shanthikumar. Approximate queueing models of dynamic job shops. *Management Science*, 31:870–887, 1985.

J. Buzen. *Queueing Network Models of Multiprogramming.* PhD thesis, Harvad University, 1971.

K. Chandy. The analysis and solutions for general queueing networks. In *6th Anaual Princeton Conference on Information Science and Systems*, 1972.

K. Chandy, U. Herzog, and L. Woo. Parametric analysis of queueing networks. *IBM Journal of Research and Development*, 19:36–42, 1975.

W. Choi and H. Shin. A real-time sequence control system for the level production of the automobile assembly line. *Computers Industriy and Engineering*, 33:769–772, 1997.

A. Conway and N. Georganas. Recal - a new efficient algorithm for the exact analysis of multiple-chain closed queueing networks. *Journal of ACM*, 33:768–791, 1986.

D. Cox. A use of complex probabilities in the theory of stochastic processes. *Proc. Cambridge Philosphical Society*, 51:313–319, 1955.

Y. Dallery. Approximate analysis of general open queueing networks with restricted capacity. *Performance Evaluation*, 11:209–222, 1990.

F. Denning and J. Buzen. The opertional analysis of queueing network models. *Computing Surveys*, 10:225–261, 1978.

R. Disney and D. Konig. Queueing networks: a survey of their random processes. *SIAM Review*, 27:335–403, 1985.

M. Dotoli and M. Fanti. Deadlock detection and avoidance strategies for automated storage and retrieval systems. *IEEE Transactions on Systems, Man and Cybernetics*, 37:541–552, 2007.

M. Dotoli and M. P. Fanti. A coloured petri net models for automated storage and retrieval systems serviced by rail-guided vehicles: a control perspective. *International Journal of Computer Integrated Manufacturing*, 18:122–136, 2005.

M. Dotoli, M. Fanti, and G. Iacobellis. Comparing deadlock detection and avoidance policies in automated storage and retrieval systems. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, 2004.

T. Duc and R. De Koster. Travel time estimation and order batching in a 2-block warehouse. *European Journal of Operational Research*, 176:374–388, 2007.

P. Egbelu. Framework for dynamic positioning of storage/retrieval machines in an automated storage/retrieval system. *International Journal of Production Research*, 29:17–37, 1991.

P. Egbelu and C. Wu. A comparison of dwell point rules in an automated storage/retrieval system. *International Journal of Production Research*, 31:2515–2530, 1993.

B. Ekren, S. Heragu, and C. Krishnamurthy, A.and Malmborg. Simulation based experimental design to identify factors affecting performance of avs/rs. *Computers Industriy and Engineering*, 58:175–185, 2010.

A. Erlang. Solution of some problems in the theory of probabilities of some significance in autmatic telephone exchanges. *Post Office Eletrical Engineering's Journal*, 10: 189–197, 1917.

M. Fanti, B. Maione, and B. Turchiano. Event-based feedback control for deadlock avoidance in flexible production systems. *IEEE Transactions on Robotics and Automation*, 13:347–363, 1997.

M. P. Fanti. Event-based controller to avoid deadlock and collisons in zone-control agvs. *International Journal of Production Research*, 40:1453–1478, 2002.

M. Fukunari and C. J. Malmborg. An efficient cycle time model for autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 46:3167–3184, 2007.

M. Fukunari, K. P. Bennett, and C. J. Malmborg. Decision-tree learning in dwell point policies in autonomous vehicle storage and retrieval systems (avsrs). In *2004 International Conference on Machine Learning and Applications, ICMLA '04*, 2004.

K. Furmans, M. Schleyer, and F. Schonung. A case for material handling systems, specialized on handling small quantities. In *The 10th International Material Handling Research Colloquium*, 2008.

E. Gelenbe and G. Pujolle. *Introduction to Queueing Networks.* Wiley, Chichester, 1987.

C. Glassey and M. Resende. Closed-loop job release control for vlsi circuit manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 1:36–46, 1988.

W. Gordon and G. Newell. Closed queueing systems with exponential servers. *Operations Research*, 15:254–265, 1967.

S. Graves. A review of production schedule. *Operations Research*, 29:646–675, 1981.

S. Graves, W. Hausman, and L. Schwarz. Storage-retrieval interleaving in automatic warehousing systems. *Management Science*, 23:935–946, 1977.

W. Hausman, L. Schwarz, and S. Graves. Optimal storage assignment in automatic warehousing systems. *Management Science*, 22:629–638, 1976.

Y. Hu, S. Huang, C. Chen, W. Hsu, A. Toh, C. Loh, and T. Song. Travel time analysis of a new automated storage and retrieval system. *Computers & Operations Research*, 32:1515–1544, 2005.

H. Hwang and C. Ko. A study on multi-aisle system served by a single storage/retrieval machine. *International Journal of Production Research*, 26:1727–1737, 1988.

H. Hwang and J. Lim. Deriving an optimal dwell point of the storage/retrieval machine in an automated storage/retrieval system. *International Journal of Production Research*, 31:2591–2602, 1993.

R. Inman. Asrs sizing for recreating automative assembly sequences. *International Journal of Production Research*, 41:847–863, 2003.

J. Jackson. Jobshop-like queueing systems. *Management Science*, 10:131–142, 1963.

J. Jia and S. Heragu. Solving semi-open queueing networks. *Operations Research*, 2: 1–11, 2009.

F. Kelly. Networks of queues with customers of different typs. *Journal of Applied Probability*, 12:542–554, 1975.

H. Kobayashi. Application of the diffusion approximation to queueing networks. *Journal of ACM*, 21:316–328, 1974.

D. Kouvatsos. Maximum entropy methods for general queueing networks. In *International Conference on Modeling Techniques and Tools for Performance Analysis*, 1985.

W. Krzesinski. Multiclass queueing networks with state-dependent routing. *Performance Evaluation*, 7:125–143, 1987.

W. Krzesinski, P. Teunissen, and P. Kritzinger. Mean value analysis for queue dependent servers in mixed multiclass queueing networks. Technical report, University of Stellenbosch, South Africa, 1981.

P. Kuehn. Approximate analysis of general queueing network by decomposition. *IEEE Transactions on Communications*, 27:113–126, 1979.

P. Kuo, A. Krishnamurthy, and C. J. Malmborg. Queuing network models for unit load storage and retrieval systems using autonomous vehicle technology and resource conserving policies. *International Journal of Intelligent Systems and Technology*, (in press), 2006.

P. Kuo, A. Krishnamurthy, and C. J. Malmborg. Design models for unit load storage/retrieval system using autonomous vehicle techniques and resource conserving storage and dwell point policies. *Applied Mathematical Modeling*, (in press), 2007.

M. Lambrecht, P. Ivens, and N. Vandaele. Aclips: a capacity and lead time integrated procedure for scheduling. *Management Science*, 44:1548–1561, 1998.

E. Lazowska and J. Zahorjan. Multiple class memory constrained queueing networks. In *International Conference on Measurement and Modeling of Computer Systems*, 1982.

S. Lee, R. Souza, and E. Ong. Simulation modelling of a narrow aisle automated storage and retrieval system (as/rs) serviced by rail-guided vehicles. *Computers in Industry*, 30:241–253, 1996.

A. Lemoine. Networks of queues - a survey of equilibrium analysis. *Management Science*, 24:464–481, 1977.

S. Lin and H. Wang. Modeling an automated storage and retrieval system using petri nets. *International Journal of Production Research*, 33:237–260, 1995.

J. Little. A proof of the queueing formula $l = \lambda w$. *Operations Research*, 9:383–387, 1961.

R. Magarajan, J. Kurose, and D. Towsley. Approximation techniques for computing packet loss infinite-buffered voice multiplexers. *IEEE Journal on Selected Areas in Communications*, 9:368–377, 1991.

C. Malmborg and K. Altassan. Analysis of storage assignment policies in less than unit load warehousing systems. *International Journal of Production Research*, 36: 3459–3475, 1998.

C. J. Malmborg. Conceptualizing tools for autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 40:1807–1822, 2002.

C. J. Malmborg. Interleaving dynamics in autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 41:1057–1069, 2003.

M. Mansuri. Cycle-time computation, and dedicatated storage assignment, for as/r systems. *Computers Industry Engineering*, 33:307–310, 1997.

R. Marie. Calculating equilibrium probabilities for $\lambda(n)/c_k/1/n$ queues. *ACM Sigmetrics Performance Evaluation Review*, 9:117–125, 1980.

J. O. Mayhugh and R. E. McCormick. Steady-state solution of the queu m/ek/r. *Management Science*, 14:692–712, 1968.

G. Meng and S. Heragu. Batch size modeling in a multi-item, discrete manufacturing system via an open queueing network. *IIE Transactions*, 36:743–753, 2004.

G. Meng, S. Heragu, and H. Zijm. Reconfigurable layout problem. *International Journal of Production Research*, 42:4709–4729, 2004.

D. Neuse and K. Chandy. Scat: a heuristic algorithm for queueing network models of cumputing systems. *ACM Sigmetrics Performance Evaluation Review*, 10:59–79, 1981.

M. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. John Hopkins University Press, 1981.

M. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Courier Dover Publications, 1995.

S. Panwalker and W. Iskander. A survey of scheduling rules. *Operations Research*, 25:45–61, 1977.

B. Park. An optimal dwell point policy for automated storage/retrieval systems with uniformly distributed, rectangular racks. *International Journal of Production Research*, 39:1469–1480, 2001.

H. Perros, Y. Dallery, and G. Pujolle. Analysis of a queueing network model with class dependent window flow control. In *IEEE INFOCOM*, 1992.

B. Peters, J. Smith, and T. Hale. Closed form models for determing the optimal dwell point location in automated storage and retrieval systems. *International Journal of Production Research*, 34:1757–1771, 1996.

I. Potrc, T. Lerher, J. Kramberger, and M. Sraml. Simulation model of multi-shuttle automated storage and retrieval systems. *Journal of Materials Processing Technology*, 157-158:236–244, 2004.

G. Pujolle and W. Ai. A solution for multiserver and multiclass open queueing networks. *INFOR*, 24:221–230, 1986.

V. Ramaswami. A stable recursion for the steady state vector in markov chains of m/g/1-type. *Stochastic Models*, 4:183–189, 1988.

M. Reiser. Mean-value analysis and convolution method for queue-dependent servers in closed queueing networks. *Performance Evaluation*, 1:7–18, 1981.

M. Reiser and H. Kobayashi. Accuracy of diffusion approximation for some queueing systems. *IBM Journal of Research and Development*, 18:110–124, 1974.

M. Reiser and H. Kobayashi. Queueing networks with multiple closed chains: theory and computational algorithms. *IBM Journal of Research and Development*, 19: 283–294, 1975.

M. Reiser and S. Lavenberg. Mean-value analysis of closed multichain queueing networks. *Journal of ACM*, 27:313–322, 1980.

M. Rosenblatt and Y. Roll. Warehouse design with storage policy considerations. *International Journal of Production Research*, 22:809–821, 1984.

M. Rosenblatt and Y. Roll. Warehouse capacity in a stochastic environment. *International Journal of Production Research*, 26:1847–1851, 1988.

Z. Sari, S. Grasman, and N. Ghouali. Impact of pickup/delivery stations and restoring conveyor locations on retrieval time models of flow-back automated storage and retrieval systems. *Production Planning & Control*, 18:105–116, 2007.

C. Sauer and K. Chandy. *Computer Systems Performance Modeling.* Prentice-Hall, 1981.

P. Schweitzer. *Aggregation Methods for Large Markov Chains.* Mathematical computer performance and reliability, 1984.

M. Segal and W. Whitt. A queueing network analyzer for manufacturing. In *The 12th International Teletraffic Congress*, 1989.

J. Shanthikumar and J. Buzacott. Open queueing network models of dynamic jobshops. *International Journal of Production Research*, 19:255–266, 1981.

M. Srinivasan and S. Heragu. Analysis of manufacturing systems via semi-oqns. *Department Working Paper, Department of Industrial Engineering, University of Lousiville, KY, 40292*, 2008.

R. Suri. Robustness of queueing network formulas. *Journal of ACM*, 30, 1983.

TGW-ERMANCO. Magnus unit load as/rs. Technical report, 2007.

A. Thomasian and P. Bay. Analysis of queueing network models with population size constraints and delayed blocked customers. In *ACM Sigmetrics Conference*, 1984.

D. Towsley. Queueing network models with state-dependent routing. *Journal of ACM*, 27:323–337, 1980.

R. Walstra. Nonexponential networks of queues: a maximum entropy analysis. *ACM Sigmetrics Performance Evaluation Review*, 13:27–37, 1985.

U. Wen, D. Chang, and S. Chen. The impact of acceleration/deceleration on travel-time models in class-based automated s/r systems. *IIE Transactions*, 33:599–608, 2001.

W. Whitt. The queueing network analyzer. *The Bell System Technical Journal*, 62: 2779–2815, 1983.

D. Yao and J. Buzacott. Modeling a class of state-dependent routing in flexible manufacturing systems. *Annals of Operations Research*, 3:153–167, 1985.

M. Yu and R. Koster. Performance approximation and design of pick-and-pass order picking systems. *IIE Transactions*, 40:1054–1069, 2008.

J. Zahorjan and E. Wong. The solution of separable queueing network models using mean value analysis. *ACM Performance Evaluation Review*, 8:255–170, 1981.

J. Zahorjan, D. Eager, and H. Sweillam. Accuracy, speed, and convergence of approximate mean value analysis. *Performance Evaluation*, 8:255–270, 1988.

L. Zhang. *Methodological Foundations for Design Conceptualization of Autonomous Vehicle Storage & Retrieval Systems.* PhD thesis, Rensselaer Polytechnic Institute, 2008.

CURRICULUM VITAE

Xiao Cai

Department of Industrial Engineering

Speed School of Engineering, University of Louisville

Louisville, KY 40292

Phone:(502)296-7789

Education

- **B.E., Control Science and Technology**

  Huazhong University of Science and Technology

  2001 - 2005

- **M.S., Industrial Engineering**

  University of Louisville

  2006 - 2007

- **Ph.D., Industrial Engineering**

  University of Louisville

  2008 - 2010

Awards & Honors

- **Graduate Research fellowship**

  LoDI (The Logistics and Distribution Institute)

  University of Louisville

  August 2007 - present.

- **Travel Award for the INFORMS Annual Meeting**

  Department of Industrial Engineering

  University of Louisville

  November 7 - 20, Seattle, Washington (2007)

  October 12 - 15, Washington, DC (2008)

  October 11 - 14, San Diego, (2009).

- **Travel Award for the INFORMS Midwest Region Meeting**

  Department of Industrial Engineering

  University of Louisville

  August 24 - 25, Evanston, Illinois (2007).

- **Graduate Research scholarship**

  Department of Industrial Engineering

  University of Louisville

  August 2006 - July 2007.

- **First Prize Scholarship Award**

  Department of Control Science and Technology

  Huazhong University of Science and Technology, China

2001 - 2005.

Professional Membership

- **Student Chapter of INFORMS at University of Louisville**

  Webmaster, 2006 - 2007

  President, 2007 - 2009

Professional Experiences

- **Research Fellow**

  LoDI, University of Louisville,

  August 2007 - present

- **Graduate Research Assistant**

  Industrial Engineering, University of Louisville,

  August 2006 - August 2007

- **Research Assistant**

  System Engineering Lab, Huazhong University of Science and Technology

  August 2005 - July 2007

- **Instructor**

  IE693-02: Computer Programming with MATLAB

  January 2010 - May 2010

IE590-02: Computer Programming in IE

August 2009 - December 2010

Presentations

- **"Web interface for conceptualization AS/RS and AVS/RS tools"**
  Cai, X., Heragu, S.S, Krishnamurthy, A. and Malmborg, C.J.,
  November 7 - 20, Seattle, Washington (2007).

Publications

- **"A New Material Handling Technology for Warehouses"**
  Heragu, S.S, Cai, X., Krishnamurthy, A. and Malmborg, C.J.,
  IE Solution, Published.

- **"Analytical Model for Analysis of Automated Warehouse Material Handling Systems"**
  Heragu, S.S, Cai, X., Krishnamurthy, A. and Malmborg, C.J.,
  International Journal of Production Research, Submitted.

- **"Open Queueing Network Model for an Autonomous Vehicle Storage/Retrieval Systems"**
  Heragu, S.S, Cai, X., Krishnamurthy, A. and Malmborg, C.J.,

2008 Industrial Engineering Conference, Vancouver, B.C., Canada, May 18-21, 2008.

- **"An Online Interface of the Conceptualization Tool of AVS/RS and AS/RS"**

  Heragu, S.S, Cai, X., Krishnamurthy, A. and Malmborg, C.J.,

  10th International Material Handling Colloquium, Dortmund, Germany, May 28-June 2, 2008.

- **"Web Interface for the Conceptualization of AVS/RS and AS/RS"**

  Heragu, S.S, Cai, X., Krishnamurthy, A. and Malmborg, C.J.,

  2008 Factory Automation and Information Management Conference, Skovde, Sweden, June 30 - July 2, 2008.

- **"Analysis of Autonomous Vehicle Storage and Retrieval System by Open Queueing Network"**

  Heragu, S.S, Cai, X., Krishnamurthy, A. and Malmborg, C.J.,

  The fifth annual IEEE Conference on Automation Science and Engineering (IEEE CASE 2009), Bangalore, India, August 22-25, 2009.

- **"Evaluate Rate Matrix in Matrix-Geometric Method for Exponential Distribution and Phase-Type Distribution"**

  Heragu, S.S, Cai, X., Krishnamurthy, A. and Malmborg, C.J.,

  International conference on value chain sustainability, Lousiville, KY, USA, October 19-21, 2009.